Techniques for Managing Grid Vulnerability and Assessing Structure

by

Jonas Kersulis

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering - Systems)
in the University of Michigan
2020

Doctoral Committee:

Professor Ian A. Hiskens, Chair
Professor Eunshin Byon
Professor Johanna Mathieu
Professor Costas Vournas, National Technical Univ. of Athens

Jonas Kersulis
kersulis@umich.edu
ORCID iD: 0000-0002-2101-241X

ACKNOWLEDGMENTS

When I first visited the University of Michigan, I met with Mads Almassalkhi, then a PhD student of Professor Hiskens. He was right to say "there's nothing like working with Ian." Without the guidance, encouragement, and example of Professor Ian Hiskens, I would not have completed this thesis. He afforded me excellent opportunities for publication, presentations, and teaching. He also arranged for me to visit other institutions, and connected me to the academic community. I cannot overstate the value of these opportunities.

Early in my studies, I visited Los Alamos National Laboratory and worked with Dr. Michael Chertkov and Dr. Scott Backhaus. Their discussions, along with Professor Dan Bienstock's knowledge of secular equations and the incisive critique of John Adams from ERCOT, aided in the transition from static to temporal instanton analysis. LANL is also where I met Miles Lubin, who inspired me to write better code and introduced me to Julia. Later on I visited the University of Queensland in Brisbane, where Dr. Tapan Saha and the UQ Power and Energy Systems group graciously hosted me. They introduced me to many impressive research projects, and to European-style bowling. More recently I worked with Professor Dan Molzahn during his time at Argonne National Laboratory. Dan's technical prowess and work ethic provide ongoing inspiration.

Many incredible people at Michigan helped me grow as a student, teacher, and researcher. Jenny Marley and Jon Martin, who joined the group before me, gave me confidence early on as we took classes and conducted research together. Professors Hiskens and Mathieu kept their power systems classes engaging with field trips and guest speakers. Professors Necmiye Ozay, Mert Pilanci, and Cynthia Finelli showed me how to teach classes on Control Systems, Digital Signal Processing, and Electronic Circuits. Professor Raj Nadakuditi gave me numerous opportunities to teach, develop curriculum, and even sing for his Computational Data Science class. I am continually inspired by his dedication to interactive teaching methods and creative data science applications.

Much of my Ann Arbor life revolved around the one-of-a-kind Escher Cooperative House, where I met many dear friends. Steve Morris has been a gym partner, counselor, and role model for years. Peter Möller joined me in Paris and hosted me in Berlin, Maja Tankoska was my Macedonian tour guide, and Alice Cox invited me to a World Cup watch party in London. Shreyas Mishra invited me to visit India, and Kevin Dunn and Courtney Ignace accompanied me there. Sky Christoph was always eager for adventure by longboard or laptop. Jack Bryant, master of Australian aphorisms, helped me prepare for my thesis proposal. GradCru, especially David Hong, and the Huron Hills Friday morning men's group built up my faith. The GradTONES gave me a wonderful a cappella musical outlet.

Thank you Rob Wilson for giving me no choice but to apply to grad schools. You were right. Thank you to my family for supporting me with every phone call, prayer, and visit. Thank you Mom and Pop for an upbringing I appreciate more each year. I love you.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF ABBREVIATIONS

SDP Semidefinite Program

OPF Optimal Power Flow

LTC Load Tap-Changing

ARIMA Autoregressive Integrated Moving Average

SARIMA Seasonal Autoregressive Integrated Moving Average

SSA Singular Spectrum Analysis

QCQP Quadratically Constrained Quadratic Program

ABSTRACT

As power systems increasingly rely on renewable power sources, generation fluctuations play a greater role in operation. These unpredictable changes shift the system operating point, potentially causing transmission lines to overheat and sag. Any attempt to anticipate line thermal constraint violations due to renewable generation shifts must address the temporal nature of temperature dynamics, as well as changing ambient conditions. An algorithm for assessing vulnerability in an operating environment should also have solution guarantees, and scale well to large systems.

A method for quantifying and responding to system vulnerability to renewable generation fluctuations is presented. In contrast to existing methods, the proposed temporal framework captures system changes and line temperature dynamics over time. The nonconvex quadratically constrained quadratic program (QCQP) associated with this temporal framework may be reliably solved via a proposed series of transformations. Case studies demonstrate the method's effectiveness for anticipating line temperature constraint violations due to small shifts in renewable generation. The method is also useful for quickly identifying optimal generator dispatch adjustments for cooling an overheated line, making it well-suited for use in power system operation.

Development and testing of the temporal deviation scanning method involves time series data and system structure. Time series data are widely available, but publicly available data are often synthesized. Well-known time series analysis techniques are used to assess whether given data are realistic. Bounds from signal processing literature are used to identify, characterize, and isolate the quantization noise that exists in many commonly-used

electric load profile datasets. Just as straightforward time series analysis can detect unrealistic data and quantization noise, so graph theory may be employed to identify unrealistic features of transmission networks. A small set of unweighted graph metrics is used on a large set of test networks to reveal unrealistic connectivity patterns in transmission grids. These structural anomalies often arise due to network reduction, and are shown to exist in multiple publicly available test networks.

The aforementioned study of system structure suggested a means of improving the performance of algorithms that solve the semidefinite relaxation of the optimal power flow problem (SDP OPF). It is well known that SDP OPF performance improves when the semidefinite constraint is decomposed along the lines of the maximal cliques of the underlying network graph. Further improvement is possible by merging some cliques together, trading off between the number of decomposed constraints and their sizes. Potential for improvement over the existing greedy clique merge algorithm is shown. A comparison of clique merge algorithms demonstrates that approximate problem size may not be the most important consideration when merging cliques.

The last subject of interest is the ubiquitous load-tap-changing (LTC) transformer, which regulates voltage in response to changes in generation and load. Unpredictable and significant changes in wind cause LTCs to tap more frequently, reducing their lifetimes. While voltage regulation at renewable sites can resolve this issue for nearby sub-transmission LTCs, upstream transmission-level LTCs must then tap more to offset the reactive power flows that result. A simple test network is used to illustrate this trade-off between transmission LTC and sub-transmission LTC tap operations as a function of wind-farm voltage regulation and device setpoints. The trade-off calls for more nuanced voltage regulation policies that balance tap operations between LTCs.

CHAPTER 1

Introduction

The unpredictability of renewable generation poses a many-faceted challenge for system operators. As generation, demand, and ambient conditions change, the system operating point moves about in its high-dimensional state space. It is important for the operator to know how far the system is from the edge of the feasible operating region, and where the closest edges are. Might some small change in the wind push the system too far in the wrong direction, causing a transmission line to overheat? If a transmission line is overheating, what is the most efficient way to relieve it? In order to be useful, any method seeking to answer these questions must capture the time-varying nature of the problem. It must also be computationally efficient, and should ideally come with a solution guarantee. When these research questions are considered in an academic environment, additional difficulties arise. Many publicly available test networks are small, represent a reduced or scrambled piece of a real power system, or were synthesized in a manner that makes them too different from real systems to be useful. Time series data, an essential component of temporal power system modeling, has issues of its own. An entire electric load dataset may be generated from one simple function. Commonly used time series datasets are stored with all values rounded to integers, resulting in surprisingly significant quantization noise. To develop and test practical methods for assessing power system vulnerability to renewable uncertainty, researchers require realistic test networks and time series data.

This dissertation introduces a deviation scanning algorithm for quickly assessing the vulnerability of a network to changes in renewable generation. The technique revolves around modeling and navigating a transmission network's feasible operating region. Each component constraint adds an edge to this region, and the system operating point can only move so far in any direction before encountering an edge. The method identifies distances and directions from the nominal operating point to all edges of the feasible region. This analysis reveals the vulnerability of the system to changes in the operating point. For each line, a non-convex quadratically-constrained quadratic program (QCQP) is used to identify the smallest change in renewable generation that would result in an unacceptably

high temperature. The constraint that forces line temperature to a limit is derived from an approximation of the standard heat balance equation for overhead conductors. Each QCQP is solved by performing a series of transformations, which reduce it to a variant of the straightforward trust region sub-problem. A system operator equipped with this analysis can steer the system towards the center of the feasible region, thereby increasing its robustness to unpredictable perturbations. The same deviation scanning algorithm can be used to compute effective re-dispatch patterns, thereby using conventional generators to relieve an over-burdened line with only minor generation adjustments. The approach scales to large systems with careful implementation, as many pieces of the numerical problem may be stored and re-used across all transmission line QCQPs.

There are several existing approaches for assessing system vulnerability to changes in generation. Those most closely related to the proposed method are the static deviation scanning techniques, which identify the smallest change in renewable generation that will cause a line to reach its steady-state limit [1, 2, 3]. This smallest change is termed the "instanton" (a name borrowed from Physics). Static instanton methods solve an optimization problem for each transmission line similar to the temporal one we propose. The objective is to minimize the change in wind (it is quadratic). Constraints include power balance and a strain constraint, which forces the line to its power flow limit. The earliest method developed used DC power flow, and simply solved the Karush-Kuhn-Tucker (KKT) conditions for each quadratic program. Subsequent research employed various power flow relaxations. One paper incorporated full AC power flow into the optimization-based framework, and took an iterative solution approach. One limitation shared by these methods is their static nature: they are unable to capture the thermal aspect of transmission line failure, or changes in generation or demand over time. Dynamic Line Rating (DLR) techniques have also been used to gauge vulnerability to renewable generation fluctuations [4, 5, 6, 7]. These approaches are complementary to the one introduced here; they typically use simulation or scenario-based analysis to assess vulnerability.

The other main thrust of this dissertation aims to distinguish realistic network structures and time series data by identifying characteristics common to physical systems and measurements. Numerous graph-theoretical techniques are applied to characterize the structure of a variety of networks, adding unprecedented breadth to classical power systems analysis. It is shown that a limited number of carefully-chosen unweighted graph properties can quickly reveal abnormal connectivity patterns, with implications for commonly-used power system analysis algorithms. A grid graph's maximal clique composition, for example, has significant performance implications for the computational performance of the Semidefinite Program (SDP) relaxation of the Optimal Power Flow (OPF) problem. Time

series characteristics are also of concern, as the temporal deviation scanning technique relies on time-varying data. The realism of publicly-available time series datasets is assessed, and a novel method for identifying and isolating quantization noise in electric load profile datasets is presented. Finally, a power system often relies on Load Tap-Changing (LTC) transformers to respond to renewable variability, and failure of these devices is a key reliability constraint. For this reason, the relationship between renewable generation voltage regulation and LTC wear is explored.

There is of course prior research related to this portion of the work as well. Researchers have used graph theory to identify system vulnerability, detect structural anomalies, and validate synthetic networks [8, 9, 10, 11, 12, 13, 14, 15]. The current work integrates and contextualizes the literature by applying many independently-proposed graph-theoretic techniques to a larger number of test networks than has previously been available. Similarly, the time series work applies several analysis techniques developed in the literature. Bringing these ideas into a common setting makes it possible to compare and combine them for greater effect. The work on improving SDP OPF performance builds on research that originally developed the clique-merge algorithm [16]. There is also precedent for study of the interaction between LTC transformers and voltage regulation at wind-farm sites [17, 18, 19, 20, 21, 22, 23]. The key advantage of the present analysis is its awareness of the trade-off between transmission and sub-transmission LTCs. This reveals the tap minimization problem to be more complicated than is suggested in the literature.

The dissertation is organized as follows. Chapter 2 introduces static instanton analysis, a prior technique that characterizes vulnerability to power injection fluctuations by computing the shortest distance from an operating point to the edge of the feasible operating region. As power systems are typically limited by the time-dependent thermal constraints of transmission lines, this limits their applicability. The static deviation scanning problem is, however, a useful testbed for power flow approximations. Before deriving a multi-time-step approach, we review the literature and introduce one straightforward technique for improving the accuracy of DC-approximate static analysis without increasing computational intensity. In Chapter 3, a temporal deviation scanning framework is developed. By linearizing the heat balance equation of a transmission line, a multi-time-step optimization problem can identify time-varying injections that will drive a transmission line to its temperature limit by the end of a specified time horizon. The novel optimization algorithm utilizes various transformations to avoid resorting to a black-box solver. In addition to identifying lines that are vulnerable to small changes in injection patterns, this technique can also identify efficient re-dispatch patterns that will drive line temperatures down. Case studies for both uses of the rapid temporal scanning method are included. Temporal

deviation scanning is also incorporated into a model predictive control (MPC) scheme to efficiently reduce line temperatures in an operating environment. The ability of the algorithm to scale to large systems is emphasized. Chapters 4, 5, and 6 address issues related to accuracy and performance of the temporal deviation scanning algorithm. In Chapter 4, time series data are analyzed with an emphasis on distinguishing realistic data from synthetic data. Quantization noise, a particularly interesting phenomenon of digitally-stored time series data, is the subject of Chapter 5. We describe a means of detecting and isolating quantization noise in load profile datasets, and provide recommendations for preventing it from arising. Development and application of the temporal scanning algorithm raises the question at the heart of Chapter 6: what is the relationship between a grid's graph structure and the realism of the network? Many graph-theoretic techniques are used to characterize the structure of numerous networks, adding breadth to prior graph-theoretic power systems analysis. It is shown that a few carefully-chosen unweighted graph properties can quickly reveal abnormal graph structures and anticipate the performance of more sophisticated analysis techniques. Chapter 7 details one such technique: the semidefinite relaxation of the optimal power flow problem (SDP OPF). It is well known that sparse decomposition of SDP OPF breaks the grid down into maximal cliques. We therefore explore the connection between clique composition and SDP OPF performance. Finally, an interesting trade-off related to load-tap-changing transformer operations is considered in Chapter 8. Renewable fluctuations induce tap operations, and voltage regulation at renewable generation sources determines how additional tap operations are apportioned among sub-transmission and distribution LTCs.

Contributions fall into two categories: the fast temporal deviation scanning technique, and a thorough treatment of network graph and time series data characteristics. The temporal deviation scanning approach provides a novel means of quantifying and responding to renewable generation variability. It is supported by advances in the understanding of time series data accuracy, power grid graph structure, performance of the semidefinite relaxation of the optimal power flow problem (SDP OPF), and LTC voltage regulation. Throughout the presentation, case studies are used to demonstrate application value, and novel visualization techniques are employed to illustrate key ideas and results.

Following is a list of publications that have proceeded from the work described in this dissertation.

- "Temperature-based instanton analysis: Identifying vulnerability in transmission networks," published with Dr. Ian Hiskens, Dr. Michael Chertkov, Dr. Scott Backhaus, and Dr. Daniel Bienstock in *IEEE PowerTech Eindhoven*, 2015 [24].

  Abstract: A time-coupled instanton method for characterizing transmission network

vulnerability to wind generation fluctuation is presented. To extend prior instanton work to multiple-time-step analysis, line constraints are specified in terms of temperature rather than current. An optimization formulation is developed to express the minimum wind forecast deviation such that at least one line is driven to its thermal limit. Results are shown for an IEEE RTS-96 system with several wind-farms.

- "Renewable voltage regulation and the transformer tapping trade-off," published with Dr. Ian Hiskens in *IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*, 2016 [25].

  Abstract: Load-tap-changing (LTC) transformers provide voltage regulation in subtransmission and distribution networks, but these expensive devices can only tap so many times before they fail. Fluctuations due to renewable energy can cause excessive tapping, accelerating aging. We address the question of how voltage regulation at renewable sources influences tapping frequency. In particular, we show that loose regulation results in excessive downstream (distribution) transformer tapping, while tight regulation causes upstream (subtransmission) tapping. We use yearlong simulations of a simple test network to generate tap trajectories and trade-off curves, illuminating the trade-off between subtransmission and distribution transformer wear. The paper concludes with a description of future work based on joint minimization of upstream and downstream tapping operations.

- "Topological graph metrics for detecting grid anomalies and improving algorithms," published with Dr. Ian Hiskens, Dr. Carleton Coffrin, and Dr. Dan Molzahn in *Power Systems Computation Conference (PSCC)*, 2018[26].

  Abstract: Power grids are naturally represented as graphs, with buses as nodes and power lines as edges. Graph theory provides many ways to measure power grid graphs, allowing researchers to characterize system structure and optimize algorithms. We apply several topological graph metrics to 33 publicly-available power grids. Results show that a straightforward, computationally inexpensive set of checks can quickly identify structural anomalies, especially when a broad set of test networks is available to establish norms. Another application of graph metrics is the characterization of computational behavior. We conclude by illustrating one compelling example: the close connection between clique analysis and semidefinite programming solver performance. These two applications demonstrate the power of purely topological graph metrics when utilized in the right settings.

- "Fast scanning for minimal wind deviations that induce temporal line overload," au-

thored with Dr. Ian Hiskens. Submission pending.

Abstract: A previously-developed method for studying a transmission network's vulnerability to wind forecast inaccuracy is expanded. The method uses optimization to find a likely wind generation pattern that brings a specified line to an unacceptably high temperature. The objective quantifies wind pattern likelihood in terms of distance from the forecast, respecting spatial and temporal correlation between wind sites and time intervals. The set of constraints enforces power balance and ensures a chosen line in the network reaches a fixed temperature by the final time step. The thermal constraint is second-order in voltage angle differences, and is based on a DC-approximate line loss formulation. Repeatedly solving the QCQP for all lines in the network yields a set of instanton candidate generation patterns, which may then be sorted by likelihood. Having described the temporal instanton QCQP and its solution, the paper turns to a discussion of implementation details. Finally, a series of numerical experiments is presented. These experiments demonstrate the effect of an instanton pattern on line temperature trajectory, the effects of wind covariance on instanton analysis, and algorithm scaling properties.

- "Quantization noise spectra in load profile datasets," authored with Dr. Ian Hiskens. Submission pending.

  Abstract: Electric load time series data are often quantized (rounded) before storage, even to integer values. This introduces quantization noise with predictable properties. This noise is well-studied in information theory, and is only negligible under certain conditions. These conditions are tested for electric load time series. Relationships between load profile dataset features, quantization step size, and quantization noise are also described and numerically validated. A process for detecting and removing quantization noise is also presented. Results demonstrate the need for caution when rounding time series data, especially after performing load disaggregation.

CHAPTER 2

Static deviation scanning

2.1 Introduction

As more wind generation is incorporated into modern power systems, wind variability plays a greater role in system operation. The unpredictability of wind far exceeds that of electric load, yet today the most common response to renewable fluctuations is reactive. To maximize the utilization of wind, operators often allow transmission lines to approach their temperature limits, then find a combination of wind curtailment and conventional generator response that relieves congestion before any line sags to an unacceptable level [27]. This approach is becoming increasingly risky. Fortunately, optimization techniques provide an alternative. The methods described in this chapter may be used to quantify and react to renewable generation variability in a static setting.

Consider a power system with several wind farms. The system operating point, described by voltage magnitudes and angles at all buses, is established according to power flow. This point exists in a high-dimensional space bounded by operating constraints. In the instantaneous setting, the region of feasible operation is a polytope formed by superposition of steady-state generation and line flow limits. System operators use generation dispatch along with load and wind forecasts to anticipate the operating point, which must lie within the feasible region (respecting all constraints). Prior to the introduction of renewable generation, the system operator's task was straightforward: conventional generation is steady, and load forecasts are accurate. The introduction of substantial wind generation increases uncertainty. In renewable-heavy power grids, the operating point is subject to greater deviations in shorter time frames. Quantifying this uncertainty and prescribing appropriate responses is one of the central topics of this thesis proposal, and doing so in the instantaneous setting is the subject of this chapter.

Suppose the demand forecast is accurate, generation dispatch does not change, and generators perform droop response to handle mismatches between total generation and

total demand. Now view the active power output of each wind farm as an optimization variable whose base value is given by a wind forecast. As wind farm outputs vary from their forecast values, the voltage magnitudes and angles describing the network operating point will move about within their high-dimensional space. Power balance constraints, line constraints, and other system constraints place boundaries on this space, characterizing the network's feasible region. If the wind forecast is accurate, the system presumably sits somewhere within this region. If the wind forecast is inaccurate, however, the true operating point will deviate from the expected one. Of all possible wind-induced shifts in the operating point, what is the most likely shift that will push the network to the edge of the feasible region?

Instanton analysis (with the term "instanton" being borrowed from Physics) answers this question using optimization. The objective is to minimize the distance between the vector of wind optimization variables and their forecast values. The most concerning scenarios are those closest to the forecast, as these are most likely to occur. A covariance matrix in the objective function accommodates situations where wind farms are likely to experience the same wind variations at the same time. The set of constraints is typical for a power grid: power balance, an angle reference, and a constraint to implement droop response (ensuring all slack is taken up). To obtain the instanton problem, one adds an additional constraint to force a line to its active power flow limit (in a particular direction). The solution to this optimization problem is an instanton candidate: it is the wind generation pattern closest to the forecast that saturates the chosen line in the chosen direction. One can then reverse the direction of the line constraint and repeat the optimization to obtain the second instanton candidate for the chosen line.

With the DC power flow approximation, the instanton optimization problem solves rapidly. It is therefore computationally feasible to repeat the optimization for all lines in the network, each in both directions. Each of the resulting instanton candidates describes the closest operating point to the forecast that will saturate at least one line constraint. The objective –distance from the forecast operating point– may be used to rank the instanton candidates. The one with the lowest objective value is closest to the wind forecast. It is the most concerning, and is designated the "instanton" in the literature. The instanton pattern and ranked list of other candidate patterns provide insight into the vulnerability of the network to wind fluctuations. They also suggest desirable and undesirable directions for the system operator to steer the network: ideally, distances to the closest instanton candidates should be increased. In the setting where a transmission line has exceeded its flow limit, the same algorithm may be used to reduce line flow.

This chapter proceeds as follows. As the instanton algorithm itself is not new, Section

2.2 provides the context of existing work. Section 2.3 presents the simplest optimization framework that may be used to solve the instanton problem. Section 2.4 describes the decoupled instanton problem, a modification of the linear model that accommodates non-unity voltage magnitudes and is therefore sensitive to line current rather than merely instantaneous active power. Section 2.6 concludes. The main contribution of this chapter is an enhancement of the DC static instanton problem: non-unity voltage magnitudes are used to derive more conservative line flow bounds, thereby increasing accuracy.

Table 2.1 contains notation that will be used throughout this chapter.

Table 2.1: Static instanton analysis notation.

$R_{p,i}$	Renewable generation at bus i in per unit; thus, R_p is a vector of optimization variables.
R_p^0	Renewable generation forecast vector.
Y_{ik}	The (i,k)th element of the admittance matrix Y, which assumes resistances throughout the network are zero.
g_{ik}	Conductance of the line between nodes i and k.
b_{ik}	Susceptance of the line between nodes i and k.
θ_k	The phase angle at bus k.
$G_{p,i}$	Conventional active power generation at bus i.
$D_{p,i}$	Active power demand at bus i.
N	The set of buses (nodes), N_r the set of renewable nodes, N_c the set of non-renewable nodes, and N_g the set of nodes with conventional generators.
k	The vector of conventional generator participation factors (the case where $k_i = 1$ corresponds to generator i taking up all slack). All elements of k are known a priori, and they must sum to 1: $\sum_{i \in N_g} k_i = 1$
α	The participation coefficient (mismatch), defined as $\alpha := \sum D - \sum \rho - \sum G_0$.
G_{p0}	The base-case conventional active power generation vector.
x_{ik}	The reactance of line (i,k).
$P_{lim,ik}$	The active power limit of line (i,k).
G	The set of edges (lines).

2.2 Prior work

Prior work has established a "distance-to-failure" intuition for instanton analysis. The DC power flow approximation was used to represent a network's feasible operating region as a set of linear constraints. These constraints form the faces of a high-dimensional polytope, the interior of which contains the feasible operating region. The boundary of the acceptable

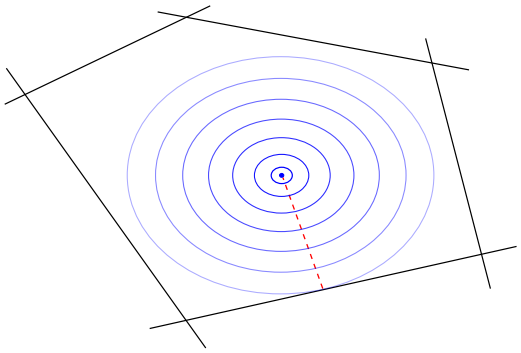


Figure 2.1: Distance from the forecast operating point to the boundary of the feasible region. The ellipses show points of equidistance from the operating point.

operating region is defined by line-flow limits. Figure 2.1 provides a two-dimensional illustration of a feasible operating region. Distance to failure is intuitively understood to be the shortest distance (smallest change) from an operating point to the boundary surface of the constraint polytope. As shown in [2], convex optimization can be used to quickly find the smallest shift in wind generation that drives the network to a chosen polytope face. Once this distance has been evaluated for every face, the collection of shifts can be sorted to determine the wind pattern corresponding to the overall smallest shift. That case, termed the "instanton", is the most likely change in wind generation that will drive the network to the brink of infeasibility.

The instantaneous DC instanton problem has been considered in [1], [2], and [3]. In [1], the problem was posed with load uncertainty (governed by a probability distribution), and controllable generators were allowed to instantaneously respond to the effects of wind variation. In [2], a more realistic setting was presented, where all controllable generators respond in an affine way to wind variability[1]. Finally, in [3] the simplest interesting version of the DC instanton problem was analyzed: one generator takes up all slack, and only line constraints are considered. The formulations of [3] and [2] have analytic solutions (satisfying the KKT conditions). It is worthwhile to go into greater detail on these previous papers.

The work in [1] formulates the DC instanton problem as a linear program. The goal is to minimize load shedding subject to a set of feasibility constraints. There are constraints on power balance, line flows, and generation capacity at each node. The set of all constraints forms the feasibility region for the system. The linear program may be viewed as a function of a demand vector d. When optimization using a certain d yields an objective value of 0

[1]The scenario where all conventional generators respond simultaneously according to pre-determined distribution coefficients is often called *distributed slack*.

(no load shedding), that demand vector is feasible. If the objective value is positive, then load must be shed, and the demand vector is infeasible. The objective changes from 0 to positive at the boundary of the feasible region. Of all operating points on the boundary, the one resulting from a demand vector closest to the forecast is the instanton. In their numerical implementation, the authors use the downhill amoeba method to solve for this output.

The authors of [1] continue their analysis of wind-induced extreme events in [2]. This chapter formulates the instanton problem in a way that permits an exact solution. Ultimately, the authors apply their formulation to the IEEE RTS-96 test network to study the effects of varying wind penetration on the solution. Loads are fixed over the optimization horizon, and renewable generation is modeled by a known (forecast) exponential distribution. Assuming no curtailment, load shedding, or network reconfiguration, the method produces a ranked list of the most problematic potential wind patterns. The instanton problem is translated to minimization of a convex function (representing deviation from forecast) over a convex set (a polytope constructed from power balance equations, line flow constraints, and generation limits). Intuitively, the algorithm does the following:

1. Construct a convex set from power balance equations, generation limits, and a single line constraint in a specified direction. (This set may be viewed as a single face of the polytope that incorporates all line flow limits simultaneously.)

2. Minimize the objective function over this set to find the point closest to the wind forecast. This is the instanton candidate associated with the chosen line constraint and flow direction.

3. Repeat the above procedure for each line in each direction to obtain an exhaustive set of instanton candidates.

4. Rank all candidates by objective value (distance from forecast operating point), and designate the one with the smallest value the instanton.

This high-level approach is the backdrop for later papers, and the remainder of this chapter. The last part of [2] demonstrates application of the instanton algorithm to the IEEE RTS-96 test network. The results provided two interesting insights into the instanton problem. First, it is possible for additional wind capacity to enhance the network's robustness to extreme wind events (in other words, adding more wind energy may make undesirable wind patterns more rare). Second, instanton wind patterns do not necessarily make intuitive sense. For example, the lines most vulnerable to wind-induced overload are not necessarily connected to nodes with wind generation. Depending on the mapping between wind injections and

operating point deviations, and the shape of the feasible space, instanton candidates are not obvious by inspection in general.

The third paper, [3], is the work of a different set of authors. The instanton component of this chapter considers both DC and nonlinear AC formulations. Solution of the DC instanton problem is achieved via a Lagrangian and the KKT conditions. To solve the AC instanton problem, an iterative method based on AC sensitivities is proposed. This method begins by viewing line current on a chosen line as a function of voltage magnitudes and angles at that line's terminal nodes. This function may be linearized at a given operating point to yield a linear relationship between changes in current and changes in voltage magnitudes and angles. Because changes in current may be related directly to changes in wind energy by a known mapping, all that remains is to solve for the change in renewable generation that would saturate the chosen line. This renewable generation change is updated in an iterative fashion with successive linearizations of the line current equation as follows: 1) Use the most recently calculated change in wind generation to update the previous vector of wind generation, 2) Solve the AC power flow at the resulting operating point, 3) Repeat the sensitivity calculation. This process typically converges after 1-2 iterations. However, because the AC power flow equations are not guaranteed to have a solution, convergence of this algorithm is not assured. For the numerical examples considered in [3], at least, the AC instanton analysis method did converge, and its results were in close agreement with those of the DC method.

AC power flow approximation research is also relevant to instanton analysis. Power system modeling is a trade-off between accuracy and numerical difficulty. The DC power flow approximation is convex and numerically well-behaved, but inaccurate when applied to systems that do not conform to its assumptions of flat voltage profile, small angle differences, and negligible line resistance. At the other end of the spectrum lies the full nonlinear AC power flow formulation, which is nonconvex but highly accurate. Between the extremes are several convex and non-convex formulations. In [28], Coffrin and Van Hentenryck describe three "Linear Programming AC" (LPAC) models for approximating power flows. These models incorporate both voltage magnitudes and phase angles, use a piecewise-linear approximation to handle the cosine in the AC power flow equations, and employ Taylor series to handle additional nonlinear terms. The LPAC models exist in three distinct contexts: hot-start, where an AC operating point is established beforehand; warm-start, where only target voltage magnitudes are known; and cold-start, where no base-case solution is available. The static deviation scanning algorithm provides a useful testbed for comparison of these approximate power system models.

2.3 Static deviation scanning

Static deviation scanning, like most power systems analysis, is easiest under DC power flow assumptions. Let all reactive power flows and line resistances be neglected, and assume all voltage magnitudes are equal to 1 pu. Suppose further a "distributed slack" model, where each conventional generator takes up slack according to a pre-defined participation coefficient. Then the instantaneous DC instanton problem may be expressed as:

$$\min \frac{1}{2} \left(R_p - R_p^0\right)^\top \Lambda \left(R_p - R_p^0\right) \tag{2.1a}$$

subject to:

$$\sum_k (Y_{ik}\theta_k) = G_i + R_{p,i} - D_i \qquad \forall i \in N \tag{2.1b}$$

$$G_i = G_i^0 + k_i \alpha \qquad \forall i \in N_g \tag{2.1c}$$

$$\theta_i - \theta_k = x_{ik} P_{lim,ik} \qquad \text{for chosen } (i,k) \in G \tag{2.1d}$$

$$\theta_i = 0 \qquad \text{where bus } i \text{ is the angle reference} \tag{2.1e}$$

$$R_i \geq 0 \qquad \forall i \in N_r \tag{2.1f}$$

$$R_i = 0 \qquad \forall i \in N \backslash N_r \tag{2.1g}$$

The objective, (2.1a), expresses the desire for small deviations by minimizing distance from the forecast operating point R_p^0. The power balance constraint (2.1b) relates active power injections to active power flows across lines. Constraint (2.1c) describes droop response. Generator G_i participates in droop response with participation factor k_i. This participation factor indicates what fraction of active power mismatch the generator will compensate for. Thus, the sum of all participation factors k_i is one. The next constraint, (2.1d), requires the solution to lie on an edge of the feasible region by constraining flow across a chosen line (i,k) to be equal to its active power limit $P_{lim,ik}$. Constraint (2.1e) defines an angle reference bus. The last two constraints describe the behavior of renewable generators: they may only produce power (not consume it), and renewable generation must be identically zero for all buses without it.

The solution of the convex optimization problem (2.1) may be found by solving its Karush-Kuhn-Tucker (KKT) conditions [29]. Let us first express the problem using linear algebra. Let the set of nodes N be partitioned into N_r, the set of nodes with renewable energy, and N_c, the set of nodes with conventional generation and demand. Sorting the

rows of the admittance matrix Y, one obtains

$$Y = \begin{bmatrix} Y_r \\ Y_c \end{bmatrix} . \tag{2.2}$$

Partitioning the remaining network parameters in this way, one can re-write the power balance constraint as follows:

$$\begin{bmatrix} Y_r \\ Y_c \end{bmatrix} \theta + \begin{bmatrix} D_r - G_r - R_p \\ D_c - G_c \end{bmatrix} = 0 \, , \tag{2.3}$$

where D_r and G_r contain all components of the demand and conventional generation vectors associated with nodes that have renewable energy, and D_c and G_c contain components associated with nodes that have no renewable energy. To capture conventional generation response, we replace G with the following affine expression:

$$G = G_0 + k\alpha \, , \tag{2.4}$$

where k is a predetermined column vector of generator participation factors which sum to one (see Eq. 6 in [30]). Each generator's droop characteristic output is given by:

$$G^i = G_0^i + k^i \alpha \tag{2.5}$$

Summing over all i, we obtain the following:

$$\sum_i G^i = G_0^{tot} + \alpha \tag{2.6}$$

So we see that α balances total generation and total load:

$$\begin{aligned} \alpha + G_0^{tot} + R_p^{tot} - D^{tot} &= 0 \\ \implies \alpha &= D^{tot} - R_p^{tot} - G_0^{tot} \end{aligned} \tag{2.7}$$

The Lagrangian term associated with α is:

$$\lambda_\alpha \left[D^{tot} - R_p^{tot} - G_0^{tot} - \alpha \right] \tag{2.8}$$

The slack bus angle reference is established as follows:

$$s_{ref}^\top \theta = 0 \, , \tag{2.9}$$

where s_{ref} is a column of zeros with a one in the ith position (bus i being the angle reference). Finally, we may saturate line (i, k) in the network using the following constraint:

$$s_{ik}^{\top}\theta - P_{lim,ik} = 0 \ , \tag{2.10}$$

where s_{ik} is a column vector with positive Y_{ik} in the ith position and negative Y_{ik} in the kth position.

The last set of constraints that must be added to our optimization formulation enforce non-negativity of all optimization variables. This is necessary because it is impossible for wind farms to act as loads. There is one constraint of the following form for each $i \in N_r$:

$$R_i \geq 0 \tag{2.11}$$

Unfortunately, inclusion of these non-negativity constraints in the final set of KKT conditions would significantly complicate them. Thus, non-negativity constraints are omitted with the understanding that solutions must be checked to ensure that all renewable injections are non-negative.

Now we have a quadratic objective function with a set of homogeneous constraints, all expressed in terms of matrices and vectors. This gives rise to the Lagrangian:

$$\begin{aligned}\mathcal{L}\left(R_p, \theta, \alpha, \lambda_r, \lambda_c, \lambda_\alpha, \lambda_{ref}, \lambda_{lim}\right) = \frac{1}{2}\left(R_p - R_p^0\right)^{\top}\Lambda\left(R_p - R_p^0\right) + \\ \lambda_r^{\top}\left[Y_r\theta + D_r - \left(G_{0,r} + k_r\alpha\right) - R_p\right] + \\ \lambda_c^{\top}\left[Y_c\theta + D_c - \left(G_{0,c} + k_c\alpha\right)\right] + \\ \lambda_\alpha\left[D^{tot} - R_p^{tot} - G_0^{tot} - \alpha\right] + \\ \lambda_{ref}(s_{ref}^{\top}\theta) + \\ \lambda_{lim}(s_{ik}^{\top}\theta - P_{lim,ik})\end{aligned} \tag{2.12}$$

The KKT conditions tell us to set all partial derivatives of the Lagrangian equal to zero:

$$\frac{\partial \mathcal{L}}{\partial R_p} = 0 = (R_p - R_p^0)^\top \Lambda - \lambda_r^\top - \lambda_\alpha \tag{2.13a}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = 0 = \begin{bmatrix} \lambda_r^\top & \lambda_c^\top \end{bmatrix} \begin{bmatrix} Y_r \\ Y_c \end{bmatrix} + \lambda_{ref} s_{ref}^\top + \lambda_{lim} s_{ik}^\top \tag{2.13b}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha} = 0 = \begin{bmatrix} \lambda_r^\top & \lambda_c^\top \end{bmatrix} \begin{bmatrix} -k_r \\ -k_c \end{bmatrix} - \lambda_\alpha \tag{2.13c}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_r} = 0 = Y_r \theta + D_r - (G_{0,r} + k_r \alpha) - R_p \tag{2.13d}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_c} = 0 = Y_c \theta + D_c - (G_{0,c} + k_c \alpha) \tag{2.13e}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_\alpha} = 0 = D^{tot} - R_p^{tot} - G_0^{tot} - \alpha \tag{2.13f}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_{ref}} = 0 = s_{ref}^\top \theta \tag{2.13g}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_{lim}} = 0 = s_{ik}^\top \theta - P_{lim,ik} \tag{2.13h}$$

This system of equations may be expressed in matrix form as follows:

$$\begin{bmatrix} \Lambda & 0 & 0 & -I & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & Y_r^\top & Y_c^\top & 0 & s_{ref} & s_{ik} \\ 0 & 0 & 0 & -k_r^\top & -k_c^\top & -1 & 0 & 0 \\ -I & Y_r & -k_r & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_c & -k_c & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{1}^\top & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & s_{ref}^\top & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s_{ik}^\top & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} R_p \\ \theta \\ \alpha \\ \lambda_r \\ \lambda_c \\ \lambda_\alpha \\ \lambda_{ref} \\ \lambda_{lim} \end{bmatrix} = \begin{bmatrix} \Lambda R_p^0 \\ 0 \\ 0 \\ G_{0,r} - D_r \\ G_{0,c} - D_c \\ G_0^{tot} - D^{tot} \\ 0 \\ P_{lim,ik} \end{bmatrix} \tag{2.14}$$

Solving this set of equations via factorization yields the instanton candidate corresponding to the chosen line (but recall that (2.14) does not include non-negativity constraints, so candidates must be checked to ensure no wind generators are consuming power).

2.4 Decoupled power flow & line current constraints

Under the DC power flow assumptions, active power flow does not vary with changes in voltage magnitude. Because reactive power flow is assumed to be zero, the DC power flow

assumes that line current is independent of changes in voltage magnitude. In reality, line current has real and imaginary components, and its magnitude depends on both. Voltage magnitudes influence current magnitudes as illustrated in Figure 2.2. Unfortunately, the expression for line line current is nonlinear, with squared terms and trigonometric functions. With a reasonable approximation, however, current magnitudes may be incorporated into static deviation scanning without increasing the difficulty of the problem. Consider a current magnitude constraint on the line between nodes i and k:

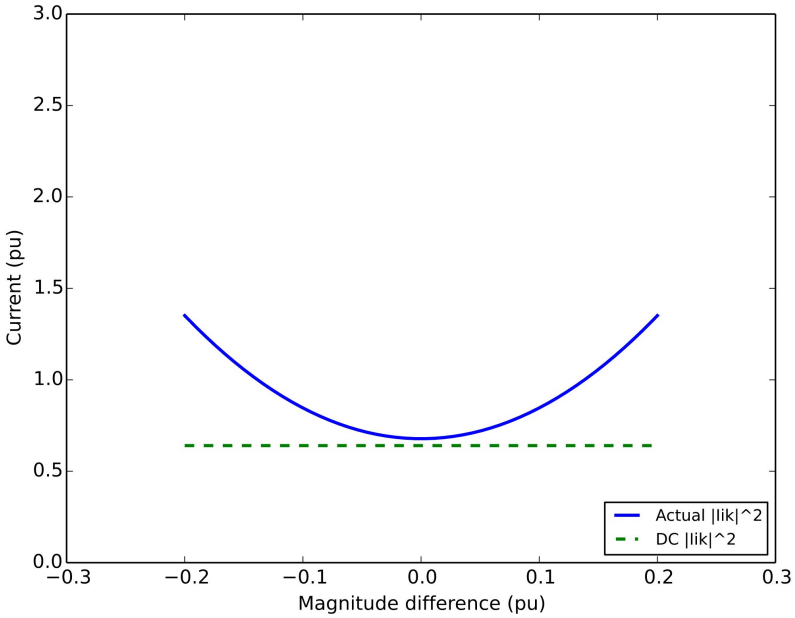


Figure 2.2: Comparison of DC and AC sensitivities to changes in voltage magnitude.

$$|I_{ik}| \leq I_{ik}^{max} \tag{2.15}$$

$$|I_{ik}|^2 \leq (I_{ik}^{max})^2 \tag{2.16}$$

$$[Re(I_{ik})]^2 + [Im(I_{ik})]^2 \leq (I_{ik}^{max})^2 \tag{2.17}$$

Next we expand the real and imaginary parts of line current I_{ik}:

$$Re(I_{ik}) = g_{ik}[Re(E_i) - Re(E_k)] - b_{ik}[Im(E_i) - Im(E_k)] \tag{2.18}$$

$$Im(I_{ik}) = b_{ik}[Re(E_i) - Re(E_k)] + g_{ik}[Im(E_i) - Im(E_k)] \tag{2.19}$$

Here we make our first assumption: the line has negligible resistance. Since $g_{ik} = 0$ by this assumption, we can simplify the expressions for real and imaginary currents:

$$Re(I_{ik}) \approxeq -b_{ik}[Im(E_i) - Im(E_k)] \tag{2.20}$$

$$Im(I_{ik}) \approxeq b_{ik}[Re(E_i) - Re(E_k)] \tag{2.21}$$

Square both expressions and add them together:

$$Re(I_{ik})^2 = b_{ik}^2[Im(E_i) - Im(E_k)]^2 \tag{2.22}$$

$$Im(I_{ik})^2 = b_{ik}^2[Re(E_i) - Re(E_k)]^2 \tag{2.23}$$

$$|I_{ik}|^2 = Re(I_{ik})^2 + Im(I_{ik})^2 = b_{ik}^2[(Im(E_i) - Im(E_k))^2 + (Re(E_i) - Re(E_k))^2] \tag{2.24}$$

Simplify:

$$\begin{aligned}
|I_{ik}|^2 &= b_{ik}^2[(Im(E_i) - Im(E_k))^2 + (Re(E_i) - Re(E_k))^2] \\
&= b_{ik}^2\Big[Re(E_i)^2 + Im(E_i)^2 + Re(V_k)^2 + Im(V_k)^2 \\
&\qquad - 2\big(Re(V_i)Re(V_k) + Im(V_i)Im(V_k)\big)\Big] \\
&= b_{ik}^2\left[V_i^2 + V_k^2 - 2V_iV_k\left(\sin\theta_i \sin\theta_k + \cos\theta_i \cos\theta_k\right)\right] \\
|I_{ik}|^2 &= b_{ik}^2\left(V_i^2 + V_k^2 - 2V_iV_k\cos\theta_{ik}\right), \quad \theta_{ik} = \theta_i - \theta_k
\end{aligned} \tag{2.25}$$

Equation (2.25) expresses the square of current magnitude on a line, assuming the line's resistance is zero (a reasonable assumption for most transmission networks). Solving (2.25) for the angle difference, we find:

$$\theta_i - \theta_k = \cos^{-1}\left(\frac{V_i^2 + V_k^2 - (|I_{ik}^{lim}|x_{ik})^2}{2V_iV_k}\right) \tag{2.26}$$

Comparing with the DC active power constraint (2.1d), we find that (2.26) takes the same form. We make the substitution in the following modified static deviation scanning formu-

lation:

$$\min \frac{1}{2}\left(R_p - R_p^0\right)^\top \left(R_p - R_p^0\right) \tag{2.27a}$$

$$\text{subject to:} \tag{2.27b}$$

$$\sum_k (Y_{ik}\theta_k) = G_i + R_{p,i} - D_i \quad \forall i \in N \tag{2.27c}$$

$$G = (1 + k\alpha)G_0 \tag{2.27d}$$

$$\theta_i - \theta_k = \cos^{-1}\left(\frac{V_i^2 + V_k^2 - (|I_{ik}^{lim}|x_{ik})^2}{2V_iV_k}\right) \quad \text{for each } (i,k) \in G \tag{2.27e}$$

$$\theta_i = 0 \quad \text{where bus } i \text{ is the angle reference} \tag{2.27f}$$

$$\sum_i k_i = 1 \tag{2.27g}$$

With a full AC power flow, the voltage magnitude values used in (2.27e) would depend on both active and reactive power flows throughout the network. We will view them as fixed instead, by treating active and reactive power as decoupled. This decoupling rests on the following assumptions: 1) active and reactive power flows are completely decoupled (the power flow Jacobian is block diagonal), 2) all wind farms operate with constant power output (not constant power factor), and 3) conventional generators act as PV buses. Because reactive power injections are constant with DC power flow, we may use these injections to establish voltage magnitudes at all nodes before beginning optimization over the decision variables (which are strictly active power injections). Since the voltage profile is no longer flat, reactive power will flow through the network, effectively de-rating line limits according to voltage magnitude differences. With the original DC formulation, line constraints were sensitive only to angle differences; under the modified formulation with decoupled power flow assumptions, these constraints are also sensitive to voltage magnitude differences. The instanton patterns obtained from the modified formulation are therefore more accurate (and more conservative) than those of the DC formulation. As long as the domain restriction on the inverse cosine is respected (as is the case during normal power system operation), the change from active power to approximate current has no adverse numerical effect on the problem.

2.5 Example

We implemented the static deviation scanning algorithm by using Julia [31] to solve both (2.1) and (2.27). To visualize key ideas behind instanton analysis, and to illustrate the

difference between active power and line current constraints, we applied both variants of instanton analysis to a modified version of the RTS-96 network. The often-used RTS-96 system consists of three interconnected areas. Our main modification is the introduction of eighteen wind-farm sites distributed unevenly. The first area has half of the wind-farms, the second has one-third, and the third area has the remainder. Wind-farm capacities are set such that wind generation is roughly 10% of total generation.

Figure 2.3 illustrates power flow corresponding to the instanton pattern identified by the algorithm (for the case where active power limits are used). In this graph layout of the modified RTS-96 system, blue circles indicate wind-farm nodes, orange circles represent conventional generators, and gray rectangles represent other nodes. Arrows between physically connected nodes indicate the direction of power flow, and arrow colors indicate how close each line is to its steady-state active power flow limit. Line 24-3 has reached its limit due to enforcement of the strain constraint (2.1d). Most lines are far from reaching their limits. In fact, under instanton conditions, only a few lines in Area 1 are strained. Note that the transmission line most vulnerable to changes in wind (line 24-3) is not directly connected to a wind-farm or conventional generator.

Performing instanton analysis with both active power limits (by solving (2.1)) and current limits (by solving (2.27)) allows us to illustrate the more conservative nature of the latter. As shown by Figure 2.4, the algorithm identifies instanton candidates with lower objective value when non-flat voltage profiles are taken into account. Voltage magnitude differences effectively de-rate the lines. This is further illustrated by Figure 2.5, which compares normalized line flows (per-unit active power as a fraction of the line's active power limits) under instanton conditions. When voltage magnitudes differ between the two buses connecting a line, decoupled power flow on that line is greater than DC-approximate active power flow. Because decoupled power flow is sensitive to voltage magnitude differences, our modified instanton algorithm is more conservative than prior algorithms that employ DC-approximate active power constraints.

2.6 Conclusion

In this chapter the static deviation scanning literature was reviewed, and an improvement to existing methods was presented. Static deviation scanning approaches provide an analytic solution that quickly provides a ranked list of renewable generation deviations which will drive at least one transmission line to its limit. The accuracy of the method is improved via decoupled power flow, which makes the method sensitive to non-flat voltage profiles. The power flow approximations and relaxations found in [32, 28, 33, 34] may also be appropri-

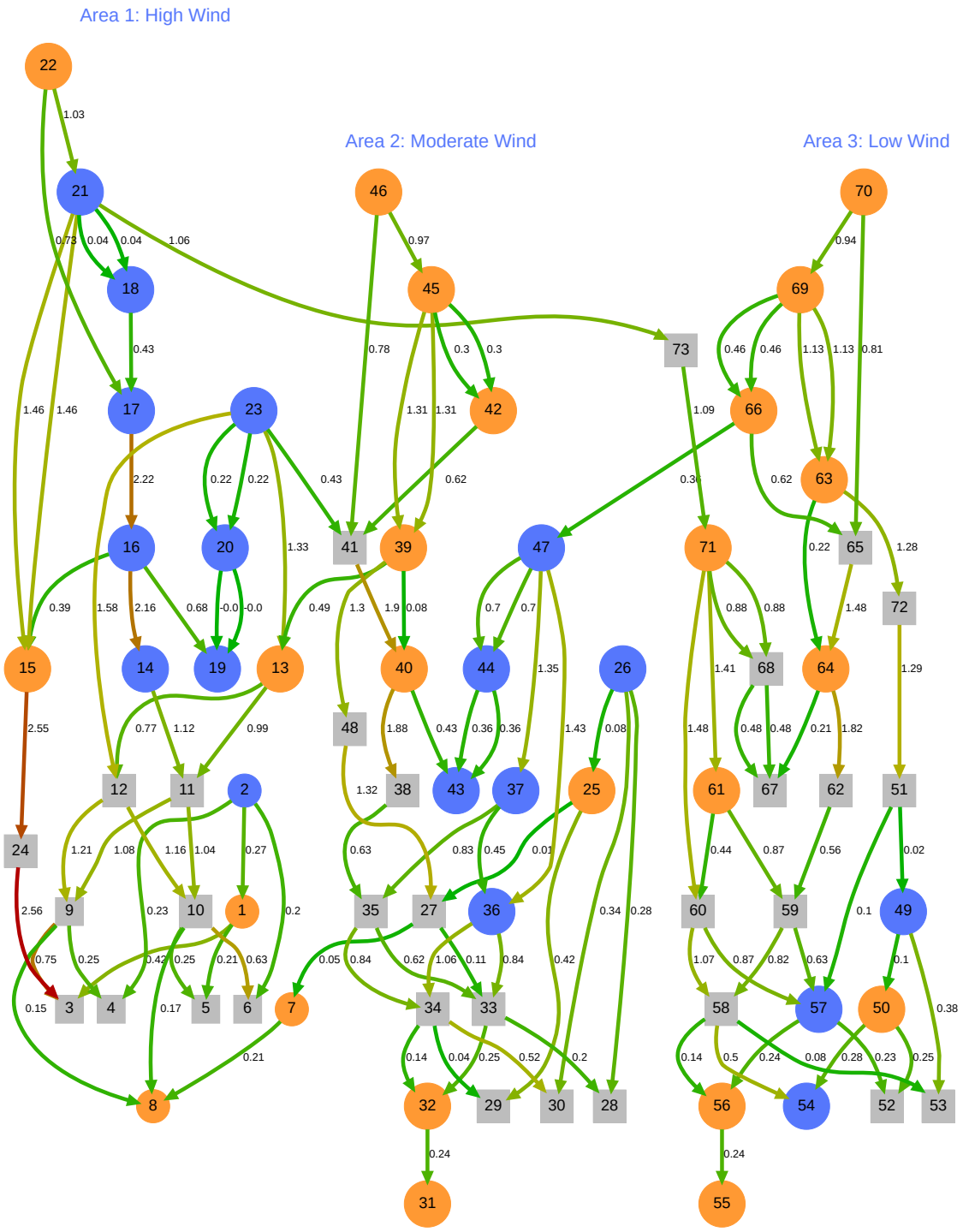


Figure 2.3: Instanton pattern identified by static deviation scanning algorithm on modified RTS-96 network. Power flow on line 3-24 is at its maximum steady-state limit.

ate for instanton analysis. Alternative wind farm control policies, such as constant power factor or constant reactive power, may also be modeled. Finally, the power flow equa-

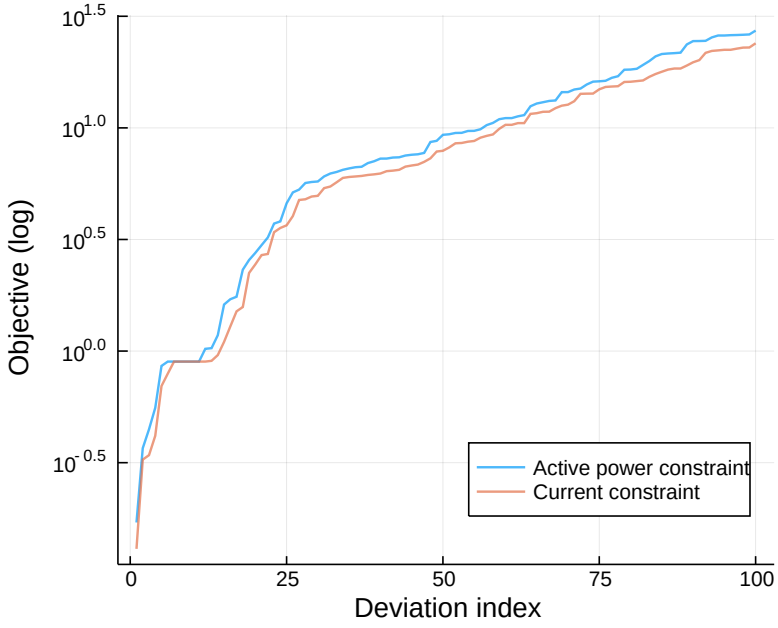


Figure 2.4: First 100 instanton objective values for analysis performed with active power limits (blue) and current limits (orange).

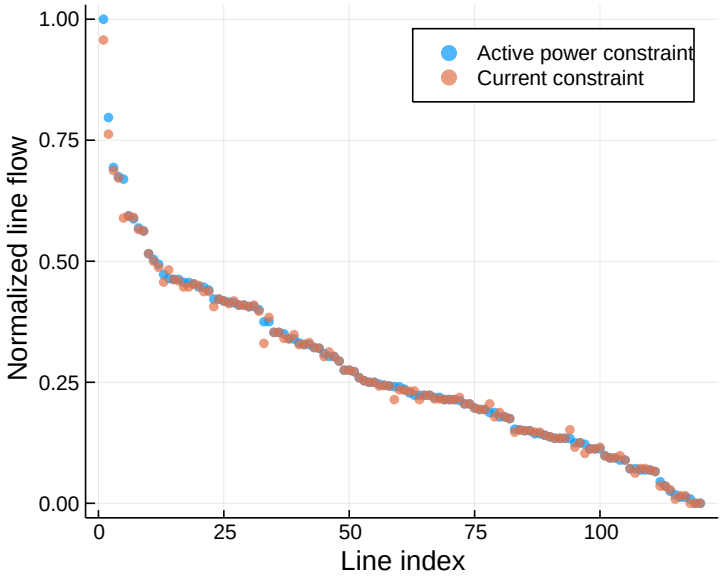


Figure 2.5: Comparison of instanton line flows for analysis performed with active power limits (blue) and current limits (orange). Each line's normalized line flow is the ratio of its per-unit active power flow to its active power flow limit.

tions may be approximated using piecewise-linear or quadratic expressions, as mentioned in Section 2.2. Many combinations of wind farm control, line flow constraints, and power

flow approximations remain untried. Comparison of various combinations would reveal the effects of these differences on instanton candidate patterns. Such a comparison could also be used to benchmark the accuracy and computational behavior of various power flow approximations. Despite the potential value of this work, there are compelling reasons to move beyond static deviation scanning.

There are two significant disadvantages to static deviation scanning methods. First, they neglect some power system components. As an example, load-tap-changing (LTC) transformers introduce discrete voltage magnitude changes over time. The effects of LTCs on static deviation scanning may be captured via unified DC power flow, but their dynamics may not be captured in static formulations. (It is worth noting here that the relationship between renewable generation voltage regulation and LTC behavior is explored in Chapter 8.) The second, and most important, limitation of static approaches is their inability to address the temporal evolution of the network. Instantaneous power limits do not accurately model transmission line sag or failure. In reality, excessive power flow causes a transmission line to accumulate heat and eventually sag to an unacceptable level; the static instanton formulation does not capture this phenomenon. Chapter 3 addresses this shortcoming with a temporal deviation scanning framework.

CHAPTER 3

Temporal deviation scanning

Chapter 2 introduced instanton analysis, a static deviation scanning technique. This method can identify, for each transmission line in a network, the smallest deviation from a forecast wind pattern that would violate its flow constraint. The resulting set of candidate deviation patterns may be used to characterize and respond to overall system vulnerability to renewable fluctuations. The key shortcoming of static methods is their inability to accommodate thermal constraints and dynamic system behavior. System operators routinely allow transmission lines to operate above their steady-state flow limits during periods of grid congestion [27]. This introduces minimal risk, provided the lines are allowed to cool before taking on too much heat. If an operator is comfortable with temporarily overloaded lines, information from static distance-to-failure analysis is too conservative to aid decision making. We address this shortcoming with the introduction of temporal deviation scanning. The new method computes the most likely wind generation pattern that would drive a specified line to its maximum allowable temperature over a chosen time frame. Wind pattern likelihood is quantified in terms of deviation from the forecast, respecting spatial and temporal correlation between renewable generation sites and time intervals. Constraints enforce power balance and ensure a chosen line in the network reaches a specified temperature by the final time step. The thermal constraint is second-order in voltage angle differences, and is based on a DC-approximate line loss formulation. The problem is therefore a Quadratically Constrained Quadratic Program (QCQP). Solving an instance of this QCQP for every line in the network yields a set of candidate deviation patterns which may then be sorted by likelihood. This set of deviation patterns can either characterize system vulnerability to renewable generation fluctuations (if these fluctuations are decision variables), or identify optimal conventional generator re-dispatch patterns for line cooling (if conventional generator outputs are decision variables). In this chapter we develop the temporal instanton QCQP formulation, present an efficient solution method, and discusses implementation details including algorithm scaling properties. The case studies in Section 3.5 illustrate the

two primary ways the algorithm may be used, as well as an example of how it might be incorporated into a real-time operating environment via model predictive control (MPC).

As wind grows into a major transmission-scale energy source, system operators frequently encounter wind-induced network congestion [35]. Operators use wind forecast data to steer the network, but wind forecasts are often less accurate than generation and demand predictions [36]. This raises the possibility that forecast deviations may compound across a collection of wind-farms to overload transmission lines. Operators need to know which lines are most vulnerable to such events. The temporal deviation scanning algorithm proposed in this chapter builds on the "distance to failure" static deviation scanning concept introduced in [37] and [2], and presented in detail in Chapter 2. These prior algorithms determine vulnerability in a static sense – they find the smallest steady-state wind generation shift that causes a line to operate at its rated power/current limit. This approach uncovers hidden grid vulnerability, but may lead to overly-conservative decision making. Temporal deviation scanning analysis, the focus of this chapter, overcomes this limitation by modeling the dynamic evolution of line temperature over a multiple time-step horizon and enforcing a temperature (rather than power/current) limit. We expand on the temporal method introduced in [24]. By modeling line temperature over an appropriate time horizon, the method discovers multi-time-step wind patterns that are both likely to occur and sure to bring at least one line in the network to its temperature limit. The new algorithm is able to quickly identify wind patterns that induce line-overload vulnerabilities, and rank them according to likelihood.

The field of Dynamic Line Rating (DLR) research has goals and perspectives similar to our own, though the technical approach of DLR differs. DLR refers to any transmission line model or limit that varies over time. Transmission lines are limited in reality by temperature and sag requirements; these phenomena are not captured by static ratings. There are three DLR scheme sub-groups pertinent to the temporal deviation scanning method. The first is comprised of competing methods, which share our aims of integrating renewable generation and characterizing system vulnerability. These methods may be used to validate or complement our own. DLR is used for integrating wind energy in [4], and for assessing reliability in [5]. Weather data and time-varying conductor resistance are incorporated into a DLR-equipped optimal power flow formulation in [38]. An approximate heat balance equation is integrated into the security-constrained unit commitment problem in the multi-period formulation of [6]. The method in [7] uses DLR and balances the benefits of increased transmission line utilization against the reserve costs necessary to compensate for forecasting error. The second group of schemes uses DLR to estimate and forecast line temperatures in real time. These methods may be used in the operating environment in

conjunction with our own. For example, initial line temperatures may be set by these methods rather than our own (recall that we compute the asymptotic value of our approximate heat balance equation). These methods may also anticipate line temperature increases and trigger a temporal deviation scan to identify potential generator re-dispatch patterns. Line temperatures may be estimated with reasonable accuracy using weather data alone [39], but of course physical measurements enhance accuracy. The comparison of real-time line monitoring devices in [40] identifies effective hardware for anticipating contingencies due to extreme line temperatures. A Kalman filter approach may be applied to such hardware measurements; for example, [41] uses sag and temperature measurements to predict future line temperature. The third group of pertinent DLR schemes compute transmission line temperature limits. These limits may be used directly in the deviation scanning algorithm. See [42] for a comparison of DLR line rating systems, and [43] for a detailed example of one such scheme.

The remainder of this chapter is organized as follows. Section 3.1 describes the three models involved in our algorithm. Section 3.2 composes these models into a quadratically-constrained quadratic program (QCQP). An efficient solution process for this QCQP is described in Section 3.3, while implementation details are discussed in Section 3.4. Section 3.5 demonstrates application of the temporal deviation scanning method with a detailed case study of two scenarios. We conclude with Section 3.6.

3.1 Models

The proposed algorithm integrates three physical phenomena. Electrical interactions are described by a network model, transmission line temperature is based on a heat balance model, and wind forecast error is quantified via a statistical model.

3.1.1 Network Model

The electrical network is modelled using the standard DC power flow [44], which assumes a flat voltage profile, negligible line resistance (though resistance values are used to calculate line temperatures), and a linear approximation of the sine function. Distributed slack is employed to ensure the mismatch between scheduled power generation and demand at any time is allocated to the generators according to their participation factors.

The network model consists of n_b buses, denoted by the set $\mathcal{B}$, of which a subset $\mathcal{W}$ of n_r buses host wind-farms. The set of transmission lines is denoted $\mathcal{E}$. The complete network model is provided in Section 3.2.

3.1.2 Transmission Line Heating

A simplified relationship between power flow and line temperature is key to achieving a tractable algorithm. Starting with IEEE standard [45], an approximate relationship is derived between a line's temperature at a time t_k and voltage angle differences at all previous time-steps $t_0, \ldots, t_{k-1}$. A summary of this relation follows, based on the derivation presented in Appendix 3.7.

Consider a time horizon consisting of n_t equal intervals, each of length τ. Let $\mathcal{T} = \{0, \ldots, n_t - 1\}$ and define $t_k = t_0 + \tau k$. Power flow quantities are computed at each time-step $t_k, k \in \mathcal{T}$, but all parameters, such as line resistances and solar heating, remain constant over the horizon. Consider a transmission line that connects nodes i and j, and let that line's temperature limit be T_{lim} (in degrees Celsius). The line's temperature at time t_{n_t} can be constrained to equal this limit value by enforcing

$$\sum_{k=0}^{n_t-1} \big(\hat{\theta}_{ij}(t_k)\big)^2 = \frac{c_1}{c_3}\big(T_{\text{lim}} - c_5\big), \tag{3.1}$$

where

$$\hat{\theta}_{ij}(t_k) = \big(\theta_i(t_k) - \theta_j(t_k)\big)\sqrt{(e^{c_1\tau})^{n_t-k+1} - (e^{c_1\tau})^{n_t-k}} \tag{3.2a}$$

$$c_1 = \frac{1}{mC_p}\left[-\eta_c - 4\eta_r(T_m + 273)^3\right] \tag{3.2b}$$

$$c_3 = \frac{r_{ij}S_b}{3mC_p x_{ij}^2 L_{ij}} \tag{3.2c}$$

$$c_4 = \big[\eta_c T_a + \eta_r\big(4T_m(T_m + 273)^3 + (T_a + 273)^4 - (T_m + 273)^4\big) + q_s\big]/mC_p \tag{3.2d}$$

$$c_5 = T_l(t_0)(e^{c_1\tau})^{n_t} + \frac{c_4}{c_1}\sum_{k=0}^{n_t-1}\Big((e^{c_1\tau})^k - (e^{c_1\tau})^{k-1}\Big). \tag{3.2e}$$

In (3.2a), $\theta_i(t_k) - \theta_j(t_k)$ is the angle difference across line (i, j) at time t_k. In (3.2b), c_1 is a constant with units of s^{-1} that is established in Appendix 3.7, mC_p is the product of mass per unit length and specific heat in J/(m°C), η_c is the convective heat loss rate coefficient in W/(m°C), η_r is the radiative heat loss rate coefficient in W/(m°C^4) and T_m is the average of ambient temperature T_a and limit temperature T_{lim} (in degrees Celsius). In (3.2c), c_3 is a constant with units of °C/s, $r_{ij} + \mathbf{j}x_{ij}$ is the impedance of line (i, j) in per unit, S_b is the system MVA base and L_{ij} is the length of line (i, j), in metres. In (3.2d), c_4 is a constant with units of W/m and q_s is the solar heat gain rate in W/m. Finally, in (3.2e), c_5 is a constant and $T_l(t_0)$ is the line's initial temperature, both with units of degrees Celsius.

See Appendix 3.7 for the derivation of this model. Because this line temperature model is second-order, there is no need to consider both directions of power flow (as there is with static analysis).

3.1.3 Wind Forecast Inaccuracy

In assessing the likelihood of a deviation in wind-power production from a forecast, spatial and temporal correlations between wind-farms should be taken into account. Consider several wind farms scattered across a transmission grid, each with a forecast power output. Let the error in this forecast at each time-step be represented by a zero-mean Gaussian random variable.[1] Then the deviation pattern across all wind-farms for a single time-step takes the form of a Gaussian random vector. Elements of this vector are correlated due to spatial relationships between wind-farms. For example, if wind speed increases at one site then it is unlikely to simultaneously decrease at a neighboring site. In addition to spatial correlation, there may also be temporal relationships between wind-farms due to weather patterns moving across a region. These are modeled by introducing non-zero off-block-diagonal elements in the precision matrix. These elements are placed such that each wind deviation is correlated with its previous and next values, thereby lowering the objective value for deviation patterns that persist across time (and hence are more realistic).

Let $\vec{r} \in \mathbb{R}^{n_t n_r}$ be the vector of forecast deviations across all wind-farms and time-steps. (The first n_t elements of $\vec{r}$ contain forecast errors for the first site at times $t_k, k \in \mathcal{T}$, the second n_t are errors for the second site, and so on.) The density function for $\vec{r}$ is given by,

$$pdf(\vec{r}) = \frac{\exp\left(-\frac{1}{2}\vec{r}^{\top}\mathbf{C}^{-1}\vec{r}\right)}{\sqrt{(2\pi)^{n_t n_r}\det\mathbf{C}}}, \tag{3.3}$$

where $\mathbf{C}$ is the covariance matrix. Maximizing pdf corresponds to minimizing $\vec{r}^{\top}\mathbf{C}^{-1}\vec{r}$. Thus, a desire to maximize wind pattern likelihood may be expressed as,

$$\min\ \vec{r}^{\top}\mathbf{Q}\vec{r}, \tag{3.4}$$

where $\mathbf{Q} = \mathbf{C}^{-1}$ is the precision matrix. There are many ways to determine $\mathbf{C}$ or $\mathbf{Q}$ from historical data. In [47], maximum likelihood optimization is used to fit a set of parameters to observed data, thereby generating a sparse precision matrix. It is also straightforward to assign geographic coordinates to each wind-farm, and map distances between wind-farms were to appropriate covariance values. A third option is to compute a sample covariance

[1] For time scales shorter than roughly one hour, a Cauchy distribution is more appropriate, but forecast errors are commonly assumed to be Gaussian nonetheless [46].

matrix from time series data. In any case, it is important to update covariance data frequently. In particular, temporal correlation is highly sensitive to wind direction, which changes often. An accurate spatio-temporal precision matrix is essential for obtaining realistic results.

3.2 QCQP Formulation

The problem of interest is to determine the most likely deviation in the wind-power forecast $\vec{r}^{*}_{\tilde{i}\tilde{j}}$, across all wind-farms $\mathcal{W}$ and all time-steps $\mathcal{T}$, that will drive a particular line $(\tilde{i}, \tilde{j})$ to its temperature limit at the end of the time horizon. This can be expressed as the QCQP,

$$\vec{r}^{*}_{\tilde{i}\tilde{j}} = \underset{\vec{r}}{\text{argmin}} \quad \vec{r}^{\top}\mathbf{Q}\vec{r} \tag{3.5a}$$

subject to,

$$\sum_{k=0}^{n_t-1} \hat{\theta}_{\tilde{i}\tilde{j}}(t_k)^2 = \frac{c_1}{c_3}\big(T_{\text{lim}} - c_5\big), \tag{3.5b}$$

and $\forall k \in \mathcal{T}$,

$$\hat{\theta}_{\tilde{i}\tilde{j}}(t_k) = \big(\theta_{\tilde{i}}(t_k) - \theta_{\tilde{j}}(t_k)\big)\sqrt{(e^{c_1\tau})^{n_t-k+1} - (e^{c_1\tau})^{n_t-k}} \tag{3.5c}$$

$$\sum_{(i,j)\in\mathcal{E}} Y_{ij}\big(\theta_i(t_k) - \theta_j(t_k)\big) = G_i^0(t_k) + g_i\alpha(t_k) + \big(R_i(t_k) + \vec{r}_i(t_k)\big) - D_i(t_k), \quad \forall i \in \mathcal{W} \tag{3.5d}$$

$$\sum_{(i,j)\in\mathcal{E}} Y_{ij}\big(\theta_i(t_k) - \theta_j(t_k)\big) = G_i^0(t_k) + g_i\alpha(t_k) - D_i(t_k) \quad \forall i \in \mathcal{B}\setminus\mathcal{W} \tag{3.5e}$$

$$\theta_{ref}(t_k) = 0, \tag{3.5f}$$

where $\vec{r}_i(t_k)$ is a notationally convenient way of referencing the appropriate element of $\vec{r}$. The objective (3.5a) matches (3.4) and expresses a desire to find wind patterns that are most likely to occur (see Section 3.1.3). The first constraint (3.5b) forces the temperature of the chosen line $(\tilde{i}, \tilde{j})$ to reach T_{lim} at the final time t_{n_t}. The second constraint (3.5c) matches (3.2a) and ties auxiliary angle-difference variables to voltage angles in the network. (Section 3.1.2 provides a detailed explanation.) Power balance is enforced by the DC power flow equations (3.5d)-(3.5e), where Y_{ij} is the (i, j)-th element of the admittance matrix (which assumes zero resistance), $G_i^0(t_k)$ is scheduled conventional generation at node i and time t_k, $R_i(t_k) + \vec{r}_i(t_k)$ is the sum of the renewable generation forecast and the forecast

error for the wind-farm at node i and time t_k, and $D_i(t_k)$ is the power demand at bus i and time t_k. The term $g_i\alpha(t_k)$ implements distributed slack, with the mismatch $\alpha(t_k)$ being distributed to generators according to their participation factor g_i. Finally, the constraint (3.5f) establishes the angle reference bus.

The mathematical program (3.5) has a quadratic objective function, a set of linear constraints, and a single quadratic constraint. This QCQP can be written compactly by combining all variables into a single vector $\vec{z} = [\vec{z}_1^\top \ \vec{z}_2^\top \ \vec{z}_3^\top]^\top$ where,

- $\vec{z}_1 \in \mathbb{R}^{n_t n_r}$ is the vector of wind deviations $\vec{r}$,
- $\vec{z}_2 \in \mathbb{R}^{n_t(n_b+1)}$ groups the angles θ and mismatch α,
- $\vec{z}_3 \in \mathbb{R}^{n_t}$ is the vector of auxiliary angle difference variables $\hat{\theta}_{\tilde{i}\tilde{j}}$.

Accordingly, $\vec{z} \in \mathbb{R}^{n_t(n_b+n_r+2)}$. This results in the formulation,

$$\begin{aligned} \min \quad & \vec{z}_1^\top \mathbf{Q} \vec{z}_1 && \text{(3.6a)} \\ \text{s.t.} \quad & \vec{z}_3^\top \vec{z}_3 = c && \text{(3.6b)} \\ & \mathbf{A}\vec{z} = \vec{b}. && \text{(3.6c)} \end{aligned}$$

The objective (3.6a) is equivalent to (3.5a), while the quadratic equality constraint (3.6b) is equivalent to (3.5b) and takes the form of a norm constraint. The linear equality constraint (3.6c) represents (3.5c)-(3.5f). The matrix $\mathbf{A}$ has $n_t(n_b+2)$ rows, comprised of n_t auxiliary angle equations (3.5c), $n_t n_b$ power flow equations (3.5d)-(3.5e), and n_t angle reference equations (3.5f).

Solving (3.6) for each line in the network yields a set of candidate wind patterns $\mathcal{R} = \{\vec{r}^*_{ij}, \forall (i,j) \in \mathcal{E}\}$, where each $\vec{r}^*_{ij}$ results in the corresponding line (i,j) being heated to its thermal limit. Of these candidates, the one with the lowest objective value, $\min \mathcal{R}$, is the most likely (instanton) wind pattern.

The following section establishes a computationally efficient solution method for QCQPs of the form (3.6), based in part on work in [48].

3.3 QCQP Solution Method

QCQPs are NP-hard in general [49]. Fortunately, the QCQP of interest (3.5) belongs to the family of trust region subproblems. As shown in [48], it may be solved in polynomial time. The solution method for the particular problem structure (3.5) can be decomposed into four steps.

3.3.1 Translation

The first step is to undertake a change of variables from $\vec{z}$ to $\vec{y} = \vec{z} - \vec{z}^*$, where $\mathbf{A}\vec{z}^* = \vec{b}$. This translation transforms (3.6c) into $\mathbf{A}\vec{y} = \vec{0}$ without qualitatively altering the objective. To prevent the change from introducing a linear term into the quadratic constraint, it is required that $\vec{z}_3^* = \vec{0}$. Thus, the desired translation must satisfy,

$$\mathbf{A}\begin{bmatrix}\vec{z}_1^* \\ \vec{z}_2^* \\ \vec{0}\end{bmatrix} = \bar{\mathbf{A}}\begin{bmatrix}\vec{z}_1^* \\ \vec{z}_2^*\end{bmatrix} = \vec{b}. \tag{3.7}$$

This problem has $n_t(n_r - 1)$ degrees of freedom, and so has an infinite number of solutions when there are multiple wind-farms, $n_r > 1$. To ensure a repeatable, well-conditioned outcome, it is convenient to choose the solution $\vec{z}^*$ of (3.7) that has minimum norm. This is a straightforward linear, least-norm problem [29], with the desired point given by,

$$\begin{bmatrix}\vec{z}_1^* \\ \vec{z}_2^*\end{bmatrix} = \bar{\mathbf{A}}^\top\left(\bar{\mathbf{A}}\bar{\mathbf{A}}^\top\right)^{-1}\vec{b}. \tag{3.8}$$

After translation, (3.6) is equivalent to:

$$\begin{aligned} \min \quad & \vec{y}_1^\top \mathbf{Q}\vec{y}_1 + 2\vec{y}_1^\top \mathbf{Q}\vec{z}_1^* && \text{(3.9a)} \\ \text{s.t.} \quad & \vec{y}_3^\top \vec{y}_3 = c && \text{(3.9b)} \\ & \mathbf{A}\vec{y} = \vec{0}. && \text{(3.9c)} \end{aligned}$$

where the constant term $\vec{z}_1^{*\top}\mathbf{Q}\vec{z}_1^*$ in the objective function has been discarded.

3.3.2 Kernel Mapping

Given the form of (3.9c), any solution to (3.9) must lie in the nullspace (kernel) of $\mathbf{A}$. Provided the flow on the line of interest $(\tilde{\imath}, \tilde{\jmath})$ is sensitive[2] to changes in the power injection of at least one wind-farm, $\mathbf{A}$ will have full row rank and an $n_t n_r$-dimensional nullspace. The second step in the solution process is to introduce another change of variables $\vec{y} = \mathbf{N}\vec{x}$ where $\vec{x} \in \mathbb{R}^{n_t n_r}$, and the $n_t n_r$ columns of $\mathbf{N}$ span $\mathcal{N}(\mathbf{A})$. This is akin to a rotation, but

[2]Such sensitivities are referred to as injection shift factors [44]. Further discussion is provided in Section 3.4.1.

reduces the problem dimension to $n_t n_r$. After partitioning $\mathbf{N}$ so that,

$$\begin{bmatrix} \vec{y}_1 \\ \vec{y}_2 \\ \vec{y}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \\ \mathbf{N}_3 \end{bmatrix} \vec{x}, \tag{3.10}$$

the previous formulation (3.9) can be written in terms of $\vec{x}$ as,

$$\begin{aligned} \min \quad & \vec{x}^\top (\mathbf{N}_1^\top \mathbf{Q} \mathbf{N}_1) \vec{x} + 2\vec{x}^\top (\mathbf{N}_1^\top \mathbf{Q} \vec{z}_1^*) && \text{(3.11a)} \\ \text{s.t.} \quad & \vec{x}^\top \mathbf{N}_3^\top \mathbf{N}_3 \vec{x} = c. && \text{(3.11b)} \end{aligned}$$

All feasible solutions of (3.11) lie in the nullspace of $\mathbf{A}$, so the linear constraints (3.9c) are implicit.

Together, the steps presented in Sections 3.3.1 and 3.3.2 comprise the well-known nullspace method [29]. Efficient implementation is discussed in Section 3.4.

3.3.3 Restoring the Norm Constraint

After kernel mapping, the quadratic constraint (3.11b) is no longer a norm constraint. This can be corrected by a change of variables that is based on the singular value decomposition [50] of $\mathbf{N}_3$. Accordingly, let $\mathbf{N}_3 = \mathbf{U}\mathbf{S}\mathbf{V}^\top$ where $\mathbf{S} = \begin{bmatrix} \hat{\mathbf{S}} & \mathbf{0} \end{bmatrix}$, and $\hat{\mathbf{S}}$ is square, diagonal and has full rank n_t. Letting $\hat{\mathbf{x}} = \mathbf{V}^\top \mathbf{x}$, the constraint (3.11b) becomes,

$$\vec{x}^\top \mathbf{N}_3^\top \mathbf{N}_3 \vec{x} = \vec{x}^\top \mathbf{V}\mathbf{S}^\top \mathbf{S}\mathbf{V}^\top \vec{x} = \hat{\mathbf{x}}^\top \mathbf{S}^\top \mathbf{S} \hat{\mathbf{x}} = c. \tag{3.12}$$

Furthermore, define the scaling,

$$\begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{S}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\vec{x}}_1 \\ \hat{\vec{x}}_2 \end{bmatrix} = \mathbf{K} \hat{\vec{x}}, \tag{3.13}$$

which gives,

$$\vec{w} = \mathbf{K}\mathbf{V}^\top \vec{x}. \tag{3.14}$$

Also note from (3.13) that $\vec{w}_1 = \begin{bmatrix} \hat{\mathbf{S}} & \mathbf{0} \end{bmatrix} \hat{\vec{x}} = \mathbf{S}\hat{\vec{x}}$, so the constraint (3.12) becomes,

$$\hat{\vec{x}}^\top \mathbf{S}^\top \mathbf{S} \hat{\vec{x}} = \vec{w}_1^\top \vec{w}_1 = c. \tag{3.15}$$

Therefore, (3.11) can be rewritten in terms of $\vec{w}$ as,

$$\min \quad \vec{w}^\top \mathbf{B}\vec{w} + \vec{w}^\top \vec{b} \tag{3.16a}$$

$$\text{s.t.} \quad \vec{w}_1^\top \vec{w}_1 = c, \tag{3.16b}$$

where

$$\mathbf{B} = \mathbf{K}^{-1}\mathbf{V}^\top \mathbf{N}_1^\top \mathbf{Q}\mathbf{N}_1 \mathbf{V}\mathbf{K}^{-1}$$
$$\vec{b} = 2\mathbf{K}^{-1}\mathbf{V}^\top \mathbf{N}_1^\top \mathbf{Q}\vec{z}_1^*.$$

The manipulations in this section have restored the norm structure of the quadratic constraint.

3.3.4 Elimination of Unconstrained Variables

The next step is to eliminate the unconstrained variables $\vec{w}_2$ from the objective function (3.16a). The KKT conditions for (3.16) will be used to do so. Expanding the objective,

$$\begin{aligned} obj(\vec{w}) &= \begin{bmatrix} \vec{w}_1^\top & \vec{w}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{12}^\top & \mathbf{B}_{22} \end{bmatrix} \begin{bmatrix} \vec{w}_1 \\ \vec{w}_2 \end{bmatrix} + \begin{bmatrix} \vec{w}_1^\top & \vec{w}_2^\top \end{bmatrix} \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} \\ &= \vec{w}_1^\top \mathbf{B}_{11}\vec{w}_1 + 2\vec{w}_1^\top \mathbf{B}_{12}\vec{w}_2 + \vec{w}_2^\top \mathbf{B}_{22}\vec{w}_2 \\ &\quad + \vec{w}_1^\top \vec{b}_1 + \vec{w}_2^\top \vec{b}_2. \end{aligned} \tag{3.17}$$

and setting the partial derivative with respect to $\vec{w}_2$ equal to zero gives,

$$\begin{aligned} \frac{\partial obj}{\partial \vec{w}_2} &= 2\vec{w}_2^\top \mathbf{B}_{22} + 2\vec{w}_1^\top \mathbf{B}_{12} + \vec{b}_2^\top = \vec{0} \\ \Rightarrow \quad \vec{w}_2 &= -\mathbf{B}_{22}^{-1}\left(\mathbf{B}_{12}^\top \vec{w}_1 + \frac{1}{2}\vec{b}_2\right). \end{aligned} \tag{3.18}$$

After substitution of (3.18) into (3.17), the objective depends only on $\vec{w}_1$,

$$\begin{aligned} obj(\vec{w}_1) = \vec{w}_1^\top &\left(\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{12}^\top\right)\vec{w}_1 \\ &+ \vec{w}_1^\top(\vec{b}_1 - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\vec{b}_2) + \text{const.} \end{aligned}$$

The constant term plays no role in the minimization and can be omitted. Thus, (3.16) is equivalent to an n_t-dimensional program with a single norm constraint,

$$\min \quad \vec{w}_1^\top \hat{\mathbf{B}} \vec{w}_1 - 2\vec{w}_1^\top \hat{\vec{b}} \tag{3.19a}$$

$$\text{s.t.} \quad \vec{w}_1^\top \vec{w}_1 = c, \tag{3.19b}$$

where

$$\hat{\mathbf{B}} = \mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{12}^\top \tag{3.20a}$$

$$\hat{\vec{b}} = -\frac{1}{2}\left(\vec{b}_1 - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\vec{b}_2\right). \tag{3.20b}$$

3.3.5 Solution via Iteration

A straightforward method of solving (3.19) involves initially diagonalizing $\hat{\mathbf{B}}$ using a singular value (or equivalently eigenvalue) decomposition as described in Section 3.3.3. Subsequent analysis assumes $\hat{\mathbf{B}}$ is diagonal. Let v be the Lagrangian multiplier associated with constraint (3.19b), so the first-order optimality conditions can be written,

$$\hat{\mathbf{B}}\vec{w}_1 = v\vec{w}_1 + \hat{\vec{b}} \tag{3.21a}$$

$$\mathbf{w}_1^\top \mathbf{w}_1 = c. \tag{3.21b}$$

This set of equations is equivalent to the *secular equation*,

$$s(v) = \sum_{i=1}^{n_t} \left(\frac{\hat{\mathbf{b}}_i}{\hat{\mathbf{B}}_{i,i} - v}\right)^2 - c = 0, \tag{3.22}$$

which has between 2 and $2n_t$ solutions.[3] Obtaining all solutions would require varying v from each pole (diagonal element of $\hat{\mathbf{B}}$) in both directions. Fortunately, the proof in Appendix 3.8 shows that optimizing (3.19) is equivalent to minimizing v subject to (3.21). Because $s(v) = \mathbf{w}_1(v)^\top \mathbf{w}_1(v) - c$ is a monotonically increasing function as v varies from $-\infty$ to the smallest diagonal element of $\hat{\mathbf{B}}$, the globally optimal value of v can be obtained using the iterative formula,

$$v^{(i+1)} = v^{(i)} - 2\frac{s\left(v^{(i)}\right) + c}{s'\left(v^{(i)}\right)}\left(\sqrt{\frac{s\left(v^{(i)}\right) + c}{c}} - 1\right), \tag{3.23}$$

[3] This is apparent from the graph of a secular equation, see [51].

which is established in [51]. The value of v obtained at convergence corresponds to the optimal $\mathbf{w}_1$ in (3.19). Reversing all the changes of variables yields the optimal value for $\mathbf{z}$ in the original problem (3.6). This vector incorporates the desired $\vec{r}^{*}_{\vec{i}\vec{j}}$, the active power deviations at all wind-farms and all time-steps.

3.4 Implementation

Two implementation aspects are of particular importance: checks that rule out infeasible QCQPs and discard unrealistic results, and numerical considerations that determine computational requirements and scaling.

3.4.1 Pre- and post- analysis Checks

Pre-analysis checks are required to identify two situations that lead to infeasible temporal instanton QCQPs. Firstly, lines that have zero resistance must be excluded.[4] As indicated in (3.31), power loss will always be zero for such lines, so the decision variables will have no effect on their temperature.

The second check identifies lines with nonzero resistance whose power flow is nevertheless unaffected by changes in wind generation. Such situations are uncommon, but may arise as a result of unusual network topology and wind-farm placement. In this scenario, the decision variables once again have no influence on the line temperature, and the QCQP is infeasible. Injection shift factors can be used to identify such lines. Whenever a line flow is fully decoupled from changes in decision variables, its shift factors with respect to all wind nodes will be zero.

Even a feasible QCQP may produce physically un-realizable results. Post-analysis checks are required to address the algorithm's inability to accommodate inequality constraints. Deviation patterns must be checked to ensure that no wind injection drops below zero, as the algorithm cannot prevent this from occurring. Generator outputs must also be checked to ensure they remain within lower and upper generation limits. Finally, the objective value of each deviation pattern does not solely determine how concerned an operator should be. While some transmission lines lie within meshed areas of the network, others merely connect a single generator to a substation. Should deviation scanning highlight one of these lines, the operator may simply reduce generation at one generator to avoid any issues. Vulnerabilities identified in the meshed parts of the network, on the other hand, may require more complex responses. This suggests that the user should leverage knowl-

[4] Such lines are typically an artifact of network book-keeping, and are not a physical reality.

edge of system structure (and results of contingency analysis, if available) before acting on deviation scanning output; each line's objective value should be weighted according to the system impact of losing it. The graph-theoretical methods described in Chapter 6 may prove beneficial here.

3.4.2 Computational Burden and Algorithm Scaling

Temporal deviation scanning involves solving the QCQP (3.5) for each line in the network. Each QCQP is independent from, but nearly identical to, all others. In fact, when one line is replaced by another, the only change in the QCQP is the choice of $\theta_{\tilde{i}}$ and $\theta_{\tilde{j}}$ in the n_t rows of $\mathbf{A}$ corresponding to (3.5c). This similarity makes it advantageous to reuse data matrices and their factorizations. The independence of each QCQP from all others also makes for pleasing parallelism: total processing time grows linearly with the number of lines analyzed. Solution time for a single QCQP varies with network size, wind-site placement and algorithm design. Factors that affect scaling will be characterized by considering algorithm complexity and numerical results.

Sparsity is of utmost implementation importance; it plays a key role in both the storage requirements and numerical manipulations. The precision matrix $\mathbf{Q}$ in the objective of (3.6) is sparse if it is generated using the method of [47]. The constraint (3.6c) is similarly sparse. The benefits of sparsity carry through to concatenation, multiplication and factorization.

Matrix factorization, a relatively expensive operation, also has a significant influence on overall algorithm scaling. Four factorizations are required in the solution method of Section 3.3. Firstly, sparse Cholesky factorization can be used to compute the pseudoinverse in (3.8) and determine the min-norm translation point $\bar{z}^*$. Further time savings are possible by exploiting the minimal differences in $\bar{\mathbf{A}}$ from one line to the next [52]. A sparse QR or LU factorization is then needed to construct a basis $\mathbf{N}$ for the kernel of $\mathbf{A}$.[5] Fortunately, this relatively expensive factorization need not be repeated for each line, as only the n_t rows of $\mathbf{A}$ corresponding to (3.5c) differ from one line to the next. Ordering those rows last, $\mathbf{A}$ can be partitioned,

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix}, \tag{3.24}$$

where $\begin{bmatrix} \mathbf{A}_2 & \mathbf{I} \end{bmatrix}$ corresponds to (3.5c). Then changing from one line to another only involves

[5] The SPQR algorithm in SuiteSparse [53] is well suited for the QR-based approach. The LU approach involves column operations on the augmented matrix $\left[\begin{smallmatrix} \mathbf{A} \\ \mathbf{I} \end{smallmatrix}\right]$. It tends to be a little faster than QR but is less numerically robust.

an update to $\mathbf{A}_2$. A null basis for $\mathbf{A}$ is any matrix $\mathbf{N}$ that satisfies,

$$\mathbf{AN} = \mathbf{0}, \tag{3.25}$$

or equivalently,

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{N}_{12} \\ \mathbf{N}_3 \end{bmatrix} = \mathbf{0}, \tag{3.26}$$

where $\mathbf{N}_{12} = \left[\begin{smallmatrix} \mathbf{N}_1 \\ \mathbf{N}_2 \end{smallmatrix}\right]$ and the subscripts on $\mathbf{N}$ coincide with (3.10). This gives,

$$\mathbf{A}_1\mathbf{N}_{12} = \mathbf{0} \tag{3.27a}$$

$$\mathbf{N}_3 = -\mathbf{A}_2\mathbf{N}_{12}. \tag{3.27b}$$

Observe in (3.27a) that $\mathbf{N}_{12}$ depends on $\mathbf{A}_1$ only. It is independent of the line being analyzed, and may be computed just once, outside the line loop. For a particular line, $\mathbf{N}_3$ is obtained via matrix multiplication according to (3.27b). The third necessary factorization is the singular value decomposition required for diagonalizing the constraint, as discussed in Section 3.3.3. Because $\mathbf{N}_3$ has dimensions $n_t \times n_t n_r$, it has only n_t nonzero singular values. These may be found directly using a dense SVD algorithm, but it is more efficient to retrieve them via Arnoldi iteration.[6] The fourth and final factorization is less demanding. Block LU decomposition [55] may be used to inexpensively compute the Schur complements $\hat{\mathbf{B}}$ and $\hat{\mathbf{b}}$ in (3.20).

3.5 Case studies

Various aspects and capabilities of the temporal deviation scanning framework are best illustrated via case studies. Flexibility, a key advantage of the method, will be showcased through two scenarios. The first is in keeping with our original motivating goal: the identification of small wind forecast deviations that would cause a transmission line to overheat. The second scenario makes use of the fact that the line temperature constraint may force a decrease in temperature rather than an increase, and that decision variables may correspond to conventional generator adjustments rather than wind deviations. This scenario illustrates the method's suitability for identifying re-dispatch patterns that alleviate a thermal overload. In either of these scenarios, the framework can incorporate changing ambient conditions such as windspeed and solar radiation. It can also accommodate changes

[6] An appropriate algorithm is described in [54] and implemented in ARPACK. It is available in MATLAB, SciPy, and Julia environments through the `svds` function.

in decision variable injections from one time step to the next (for example, demand and generation dispatch may vary over the optimization period). The method can accommodate many changes that occur in power system operation, making it useful in a variety of situations. The solution guarantee and low computational complexity (along with efficient implementation, as described previously) also makes the algorithm suitable as a sub-routine in higher-level analysis. At the end of this section, we provide an example of how temporal deviation scanning may be incorporated into an online model predictive control (MPC) scheme that manages transmission line temperature over the course of a hot summer day.

The remainder of this section is organized as follows. First, an illustrative test case is developed from the IEEE 118-bus system. Then two operating scenarios, which illustrate the main uses of the deviation scanning method, are played out in this system. The first is concerned with line temperature increases due to wind fluctuations: the method optimizes over all renewable injection deviations to identify transmission lines that are likely to overheat. The second scenario uses the method to reduce the temperature of an overheated line, by efficiently re-dispatching conventional generators. Next, we characterize algorithm scaling using two tests: one where the number of decision variables is varied, and another where analysis is performed for many popular test networks. Finally, we show how this temporal re-dispatch scanning may be incorporated into an MPC framework for use in online operation. Results illustrate the effectiveness of the method in all three scenarios. Limitations are also discussed, along with suggestions for dealing with them.

3.5.1 Static test case

The IEEE 118-bus system has an appropriate size and topology for our demonstration, but various modifications and parameter estimates are required to make it suitable for the application of our method. As the original 118-bus test case has only conventional generators, we begin by converting a portion of them into wind-farms. To mimic reality, we placed all wind-farm sites together in a low-demand, loosely-meshed part of the grid. Figure 3.3 illustrates this placement using large green circles.

We also modified the distribution of electric load such that our test case consists of three distinct regions. Referring to Figure 3.3, the upper-left portion of the system represents a fairly spread out population with moderate electric load; the dense sub-network at the bottom is an urban area, highly interconnected with high demand; and the upper-right portion of the system represents a remote and relatively unpopulated area, where it is more feasible to construct wind farms. Our modifications involved increasing demand in the areas designated as more populated, and reducing it in the remote region containing wind

Table 3.1: Thermal parameters used in case study.

Parameter	Value	Units	Notes
Emissivity	0.70	-	Value in [0.23, 0.91]
Wind-line angle	90.0	Degrees	Wind perpendicular to line
Elevation	61.0	m	Average PJM elevation
Wind speed	0.61	m/s	Just over 1 mph
Solar absorptivity	0.90	-	Value in [0.23, 0.91]
Latitude	40.0	Degrees	Midwestern US
Day of year	161	-	June 10
Line azimuth	90.0	Degrees	West-to-East
Hours from noon	0.0	hr	Scenario begins at noon
Ambient temperature	40.0	C	Hot summer day

farms.

Finally, the temporal deviation scanning algorithm relies on thermal parameters, including ambient conditions and transmission line conductor properties. Because the original IEEE 118-bus test case dataset lacks this information, we developed reasonable estimates. System-wide thermal parameters were fixed with the help of the IEEE 738 standard [45]; our choices are summarized in Table 3.1. Line-specific thermal parameters depend on conductor type, bundle configuration, and length. These parameters may be estimated for each line as follows. First, the current limit is obtained by converting the steady-state line rating from per unit to amps. Next, conductor bundling is estimated from the current limit and line voltage. The conductor type is then fixed by looking up the current limit in a table of ACSR specifications. Line length is estimated from voltage, per-unit resistance, conductor per-meter resistance, and bundle configuration. Finally, initial line temperature is estimated for a particular set of power injections by computing the asymptotic value of the line temperature equation (3.36). Straightforward computer code can automate all estimation methods described here to quickly augment any test case with reasonable thermal data.

At this point, the modified 118-bus test case possesses static characteristics appropriate for application of the deviation scanning algorithm. To demonstrate the proposed method's full utility, however, grid behavior must change over time. We next develop two time-varying scenarios to illustrate two complementary operational uses of the algorithm.

3.5.2 Assessing vulnerability to renewable fluctuations

In the first scenario, demand increases gradually in the urban area of the network over the course of an hour, divided into six ten-minute intervals. This is consistent with an early

afternoon on a hot day in the US Midwest (a summer-peaking region). In this scenario, the temporal deviation scanning method readily identifies transmission lines that are likely to reach $100°\mathrm{C}$ during the hour due to small deviations from the wind generation forecast.

For this scenario, static parameters are fixed as discussed in Section 3.5.1. Conventional generation dispatch is constant during the hour (though actual output varies according to distributed slack; see Section 3.1.1). For each ten-minute interval, demand at each urban node increases by a randomly-chosen value in $[0, 0.2]$ pu. The forecast output for each wind node also changes at each interval, via addition of a value sampled from a normal distribution with mean 0 and standard deviation 1, and scaled by 1/6.[7] Finally, a slight temporal correlation, as described in Section 3.1.3, was introduced.[8]

The results of applying temporal deviation scanning to this scenario may be illustrated in a number of ways. Figure 3.1 (top subplot) illustrates temperature trajectories for all lines in the network corresponding to the optimal (smallest) wind deviation pattern. Note that a single line, between buses 23 and 24, reaches the temperature limit of $100°\mathrm{C}$. Most lines remain close to their initial steady-state temperatures. While several lines have relatively high initial temperatures, some are more sensitive to changes in wind than others. This is illustrated further by Figure 3.2, which shows the relationship between line initial temperature and optimization objective value. In keeping with intuition, there is an inverse relationship overall (hot lines are more likely to reach their temperature limits than cool ones). However, the line with the highest initial temperature does not have the lowest deviation scanning objective value. A line's sensitivity to changes in wind plays an important role, as do its length and thermal parameters.

The temperature trajectories shown in Figure 3.1 are, of course, driven by a specific deviation pattern computed by the algorithm. Figure 3.1 (bottom subplot) illustrates this pattern. Its shape is relatively constant across time steps (due to temporal correlation), and the largest-magnitude deviations are around 0.3 pu. Finally, we overlay objective values on a graph layout of the system. Figure 3.3 is a graph layout of our modified 118-bus system. Nodes are placed according to electrical coordinates, derived via multidimensional scaling from power transfer distances.[9] Wind nodes are larger and green, while other nodes are smaller and black. The width of each edge is fixed according to voltage level: thin lines represent 138 kV, medium lines 161 kV, and thick lines 345 kV. Lines are colored according to temporal deviation scanning objective value, with lower objective value corresponding

[7] The value 1/6 was chosen to ensure no wind forecast value dropped below 0.

[8] Spatial correlation may also be incorporated into the objective function matrix by adding off-diagonal coefficients *within* each time step, as mentioned in Section 3.1.3. Our scenario omits this for simplicity.

[9] The power transfer distance between two nodes is the sum of all line flows induced by a 1-unit power transfer between them; see [56] for details.

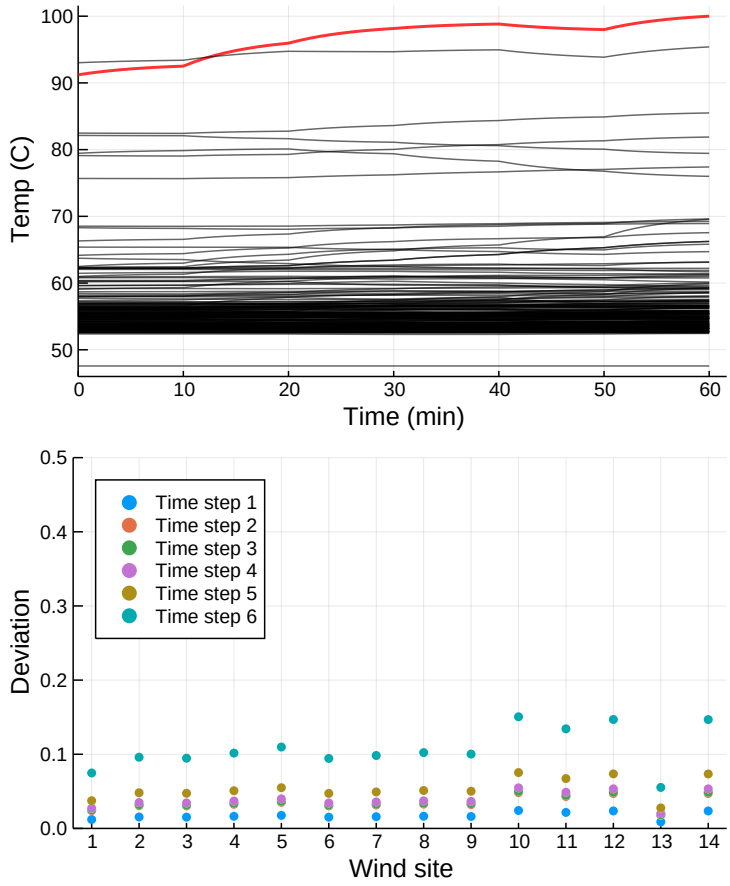


Figure 3.1: Temperature trajectories (left) and wind injection deviations (right) corresponding to deviation pattern that will bring at least one line to $100°\mathrm{C}$.

to brighter red. This illustration shows at a glance which lines are most susceptible to small changes in wind, and the power transfer distances they span. The most vulnerable line is visible near the upper-right area of the system; other vulnerable lines are scattered throughout the wind-dense and central regions.

It is important to note one limitation of the algorithm made apparent by this application. The approximate line temperature model used in temporal deviation scanning assumes that heat loss due to convection varies only with temperature (see Section 3.7). In reality, this term also varies with wind speed and direction. Thus, wind deviations identified by the algorithm at a given time step influence line thermal dynamics at that time. Accurately modeling this dependence of line temperature on wind deviations would be difficult. It would ultimately involve altering the line temperature constraint, thereby disturbing the problem structure our solution method is based on. Thus, temporal deviation scanning implicitly assumes that changes in wind are not large enough to meaningfully impact convection heat loss. This assumption is valid in the intended application setting, where wind deviations

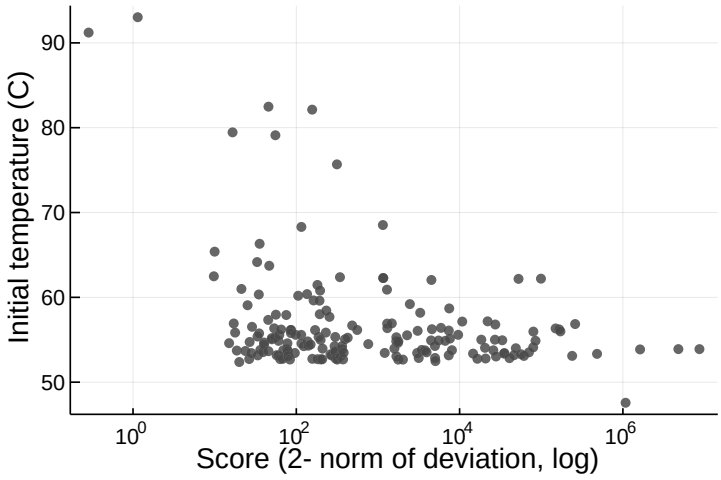


Figure 3.2: Initial temperature versus deviation scanning optimization objective value. Higher initial temperature generally corresponds to lower objective value, but shift factor sensitivities and thermal parameters are also important.

are small and persist over a short time period. As deviations (and therefore objective values) grow larger, it is worth noting that line thermal dynamics change as a result. If this is a concern in a particular setting, one solution is to make a conservative (low) windspeed estimate. This will keep convection heat loss low, and ensure that real line temperatures are always lower than simulated ones.

3.5.3 Coordinating conventional generation to reduce temperature

In the second scenario, one transmission line is already too hot, and we seek an efficient re-dispatch of conventional generators to cool it down. Suppose that urban demand is increased even further than in the previous scenario, such that one line in the network reaches $105°\mathrm{C}$. In this scenario, a system operator would like to coordinate conventional generation to reduce the temperature of this line by $10°\mathrm{C}$, to $95°\mathrm{C}$.

In this scenario, the temporal deviation scanning framework may be used to identify small changes in conventional generation that would reduce a line's temperature back to an acceptable value over a specified time frame. The structure of the problem remains the same, but now decision variables correspond to dispatchable generators rather than windfarms. Final line temperature of the overheated line is constrained to $95°\mathrm{C}$ (though any final temperature could be specified). The time horizon is also shortened to half an hour, as it is important to quickly cool the overheated line.

The results of applying the temporal deviation scanning method are shown in Figure 3.4. The line cools by the desired $10°\mathrm{C}$ over the course of half an hour (top subplot),

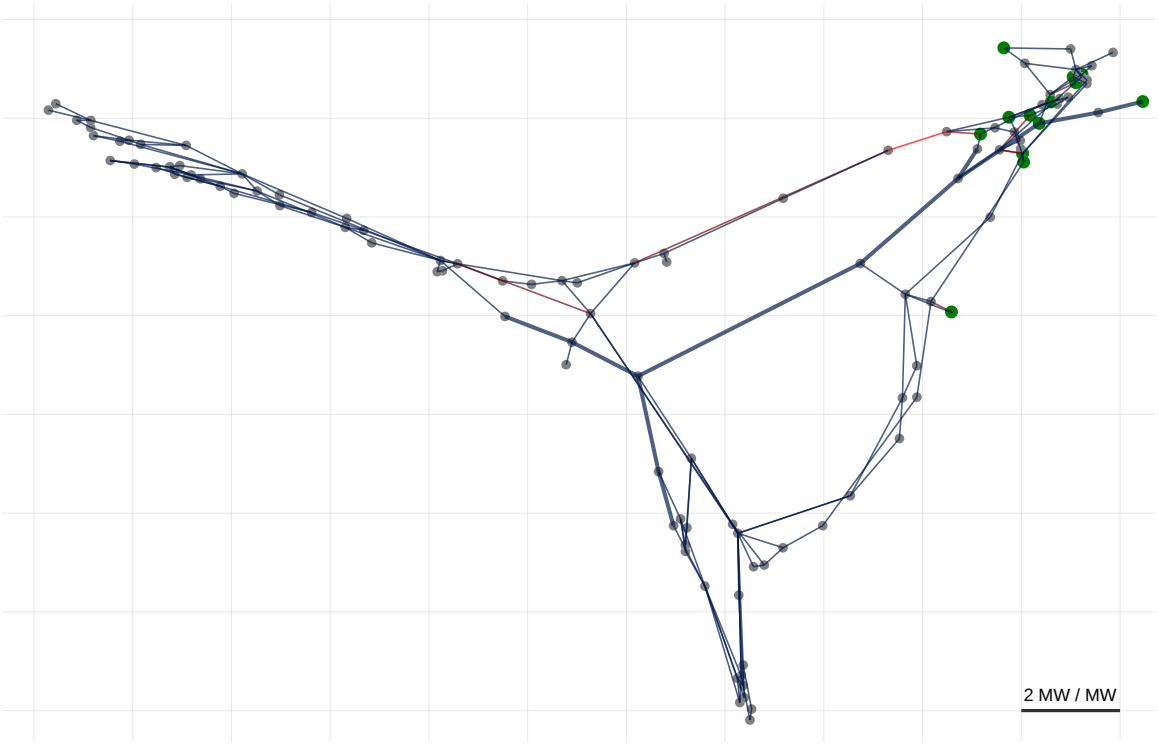


Figure 3.3: Graph layout visualization of deviation scanning objective value. Buses are placed according to electrical coordinates, derived via multidimensional scaling from power transfer distances. Wind-farm buses are larger and green. Lines deemed more vulnerable by temporal deviation scanning are redder in color.

and only relatively minor changes in dispatch are required (bottom subplot). Note that the algorithm provides no guarantee that a transmission line elsewhere in the network will not overheat due to the re-dispatch. The pattern proposed by the method should therefore be evaluated in light of resulting temperature trajectories. While it is optimal for cooling the overheated line, the pattern may cause unacceptable temperature increases for other lines.

In this setting, we used the algorithm to identify the smallest change in conventional generation, in a 2-norm sense. With a minor adjustment, and assuming the original dispatch is cost-optimal, the method may be used to compute a cost-optimal change instead. Because the objective function can include a linear term, it can incorporate second-order generator cost curve coefficients, and thereby represent the monetary cost of re-dispatch.

3.5.4 Algorithm scaling and computational complexity

Having developed a temporal deviation scanning implementation that preserves sparsity and minimizes the computational burden of factorization (see Section 3.4.2), we performed two tests to characterize algorithm scaling. Firstly, we performed analysis on a modified

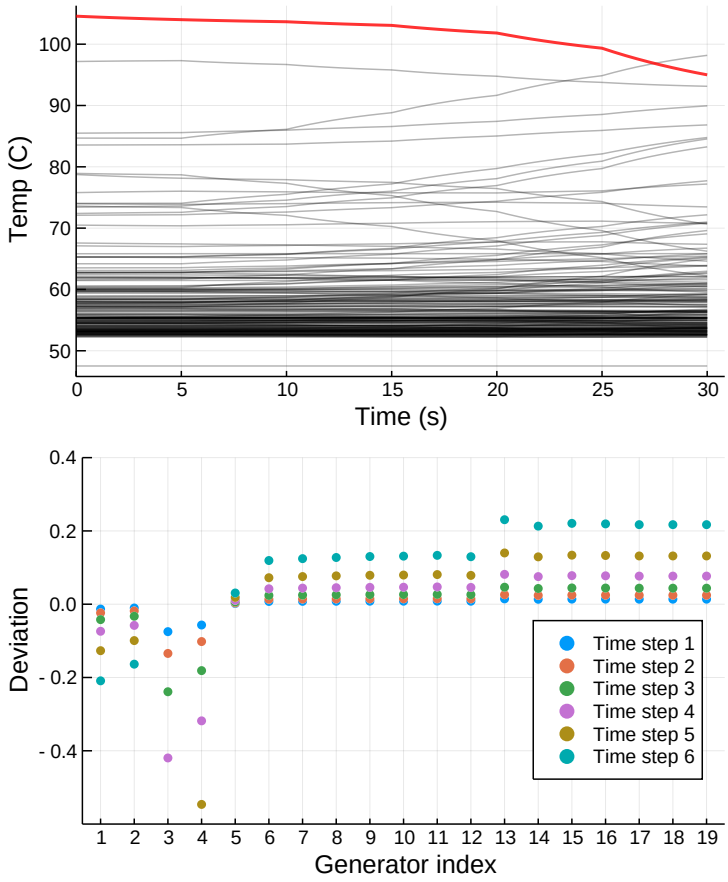


Figure 3.4: Effectiveness of temporal deviation scanning algorithm for reducing temperature of an overheated line. Temperature trajectories at top; deviation pattern at bottom.

RTS-96 network with added wind-farms. The number of wind-farm buses (and, therefore, decision variables) was varied. Using a time horizon of one hour divided into six ten-minute intervals, a complete analysis of each case was performed. Figure 3.5 illustrates the relationship between QCQP size and solution time. Secondly, wind farms were added to eighteen popular test networks (selected from PGLIB, successor to the NESTA network library [57]), and each case was analyzed. For each network, conventional generators with below-average nominal injections were replaced by wind-farms.[10] Other thermal data were chosen or estimated as in Section 3.5.1. Figure 3.6 shows average computation time for each QCQP versus network size. The data are roughly linear on a log-log scale. Figure 3.6 shows average QCQP solution time versus network size.

[10]While many wind-farm sizings and placements would work for illustrating algorithm scaling, this choice tends to avoid numerical difficulties. If there are too few wind nodes or penetration is too low, many solutions will incur absurdly high objective value.

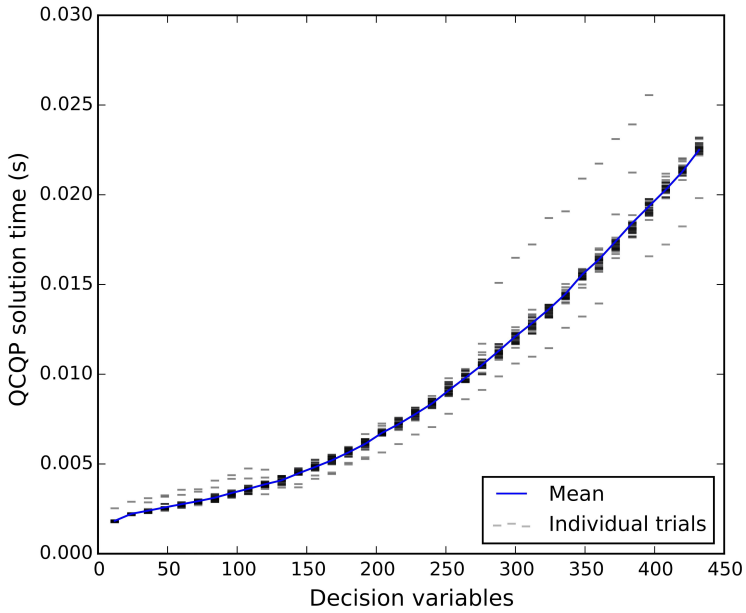


Figure 3.5: Computation time per QCQP solution versus number of decision variables. The number of wind farms is increased two at a time up to 72, with wind farms randomly placed throughout the network. The ratio of forecast wind generation to conventional generation is fixed at 0.7. Analysis was repeated thirty times to illustrate the effects of wind-farm placement on solution times.

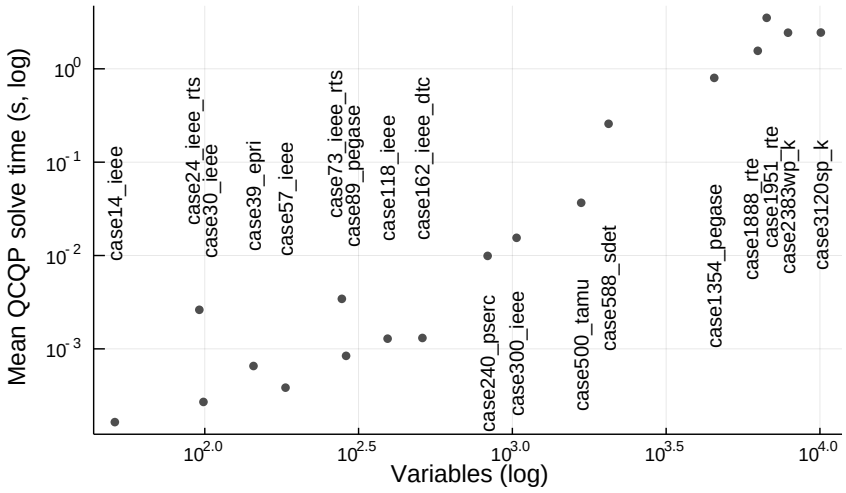


Figure 3.6: Average QCQP solution time (across a maximum of 100 lines) for eighteen popular test cases.

3.5.5 Use in model predictive control

Temporal deviation scanning is well-suited for use in model predictive control (MPC). Whereas the previous two case studies required one use of the algorithm each, an MPC framework can use the algorithm to make a sequence of efficient dispatch adjustments over

an extended period of time. MPC is a type of online process control originally developed for use in chemical plants. It is useful for controlling multi-dimensional dynamical processes in the presence of uncertainty. The main ingredients of MPC are 1) an internal dynamical model of the process, such as transmission line temperature dynamics; 2) an optimization algorithm such as temporal deviation scanning; and 3) a desired outcome (for example, keeping transmission line temperatures below a particular value). At each time step, the optimization algorithm is used to determine optimal control actions across a short-term time horizon, in order to achieve the desired outcome. Because these control actions are only optimal if the internal dynamical model is accurate and system conditions do not change, only control actions for the current time step are implemented; Future control actions are discarded. MPC then repeats the optimization at the next time step, when new information is available. In this manner, the method corrects for the effects of model inaccuracy and system uncertainty over time. Under certain conditions, MPC can even afford convergence guarantees.[11] MPC has been used effectively in power systems research [58, 59], and has been proposed specifically for controlling transmission line temperatures [60]. In this section it is shown that that an MPC framework can use temporal deviation scanning to manage line temperatures over the course of a day. As with previous case studies, we begin by developing an appropriate scenario.

3.5.5.1 Electric load

The 118-bus static test case described in Section 3.5.1 is used once more, but now we will use demand and wind data representing a hot summer day. Demand data is derived from the ERCOT dataset described in Section 4.2.3; we extracted demand data for Monday, July 10, 2017. Figure 3.7 illustrates ERCOT load data for all eight areas on this day, with each load profile normalized such that its maximum value is 1.0. To obtain a load profile for each bus in the test network, we randomly assigned one of the eight normalized profiles to each load bus, and scaled it according to the bus's nominal load. The resulting load profiles are shown in Figure 3.8. This hourly data was then interpolated to obtain 72 data points spanning 24 hours in 20-minute time intervals.

3.5.5.2 Wind generation

Wind generation data comes from NREL's WIND dataset [61]. For each of the numerous geographical sites in this dataset, NREL used site-appropriate wind turbine power curves to convert windspeed measurements to turbine power outputs. We chose 14 sites (match-

[11] Evaluation of these conditions for a temporal deviation scanning MPC is left to future work.

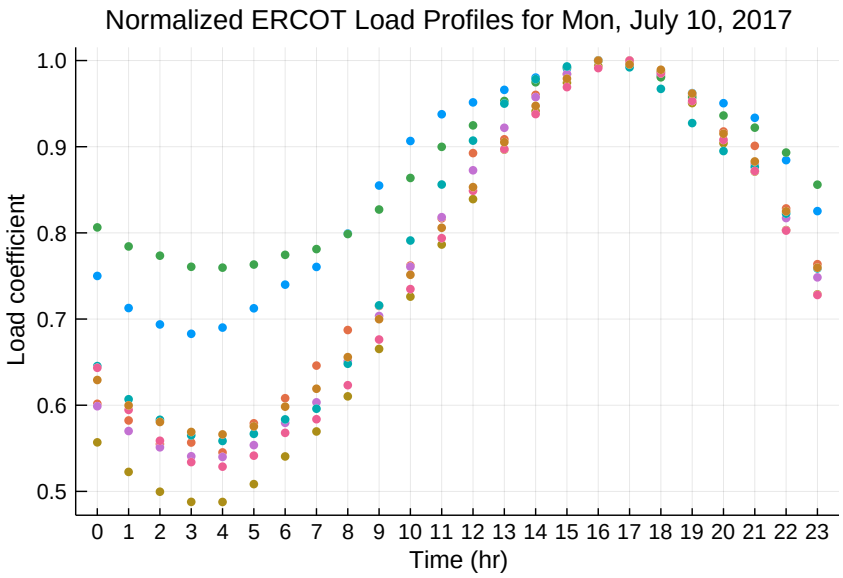


Figure 3.7: Normalized load profiles for the eight areas of ERCOT during July 10, 2017.

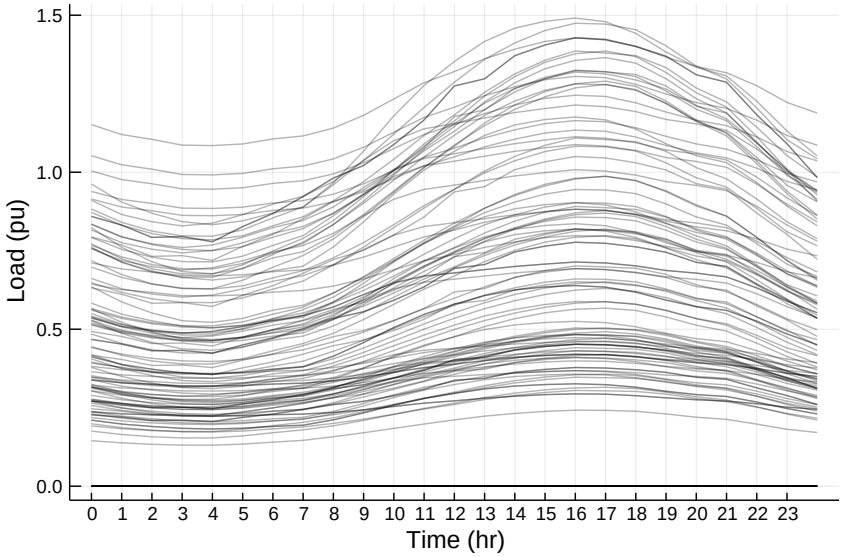


Figure 3.8: Load profiles for 118-bus test case, derived via disaggregation from the data in Figure 3.7.

ing the number of wind sites in our 118-bus test network) located within a relatively small wind-dense region near Lubbock, Texas.[12] Figure 3.9 illustrates wind turbine power outputs for these sites on July 15, 2012. As with the load data, these normalized hourly profiles are scaled according to nominal wind generation at each of the 118-bus network's wind sites. Finally, interpolation was performed to obtain 72 data points at 20-minute intervals.

[12]Chosen sites: 14135, 13647, 13993, 13996, 13850, 14176, 13587, 14131, 13710, 13847, 14177, 13789, 14132, and 13646.

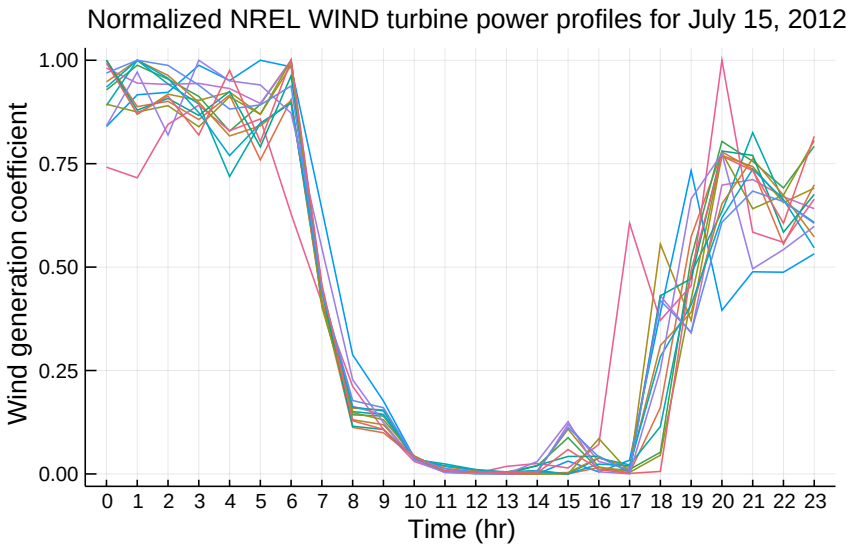


Figure 3.9: Normalized wind turbine power profiles from NREL's WIND dataset.

3.5.5.3 MPC approach

The MPC algorithm behaves as follows. At the start of each 20-minute time interval, line temperature trajectories are simulated over the MPC time horizon (we chose one hour in 20-minute intervals). Any line whose simulated temperature exceeds a chosen safe temperature ($90°\mathrm{C}$ was chosen) is added to a set. Temporal deviation scanning is then performed for this set of lines. The output is a set of optimal re-dispatch patterns, each of which will cool one line down to $90°\mathrm{C}$ by the end of the MPC time horizon. The optimal re-dispatch patterns for all lines in the set are then implemented at the current time step (deviation vectors are summed together), and the process repeats at the start of the next time interval.

This MPC framework possesses several degrees of freedom and incorporates multiple heuristics. Most importantly, implementing multiple deviation patterns at once assumes that cooling one line will not increase the temperature of another. Deviation patterns are not aligned in general; adding them together results in cancellation. Although the method performed well in the cases we have considered, it is straightforward to design a network which would limit its effectiveness. In any situation where cooling one overheated line leads to overheating another, the MPC framework will effectively take insufficient action due to deviation pattern cancellation. It is also important to take an appropriate amount of control action at each time step. Temporal deviation scanning typically recommends small control actions at early time steps, with most effort delayed until the end of the MPC time horizon. This is optimal in the sense that keeping a transmission line hot early on allows convection and radiation to have greater cooling effects, thereby reducing the necessary generation dispatch deviation. Unfortunately, in an MPC setting where only deviations for

the current time step are implemented, this behavior results in a perpetual delay of control action. The resulting MPC consistently takes insufficient action. To correct for this, we chose to implement the average deviation over all MPC horizon time steps at the current time step, effectively spreading control action evenly across the MPC time horizon. Other degrees of freedom include the choice of MPC time horizon, and whether auto-correlation is used to reduce generator dispatch changes from one time step to the next. More rigorous investigation of these degrees of freedom may prove valuable.

It is also worth noting that there may be other objectives besides keeping all transmission line temperatures below a chosen threshold. Suppose a system operator wishes to operate the network far from all thermal limits, such that any renewable generator may vary by $\pm 10\%$ without adverse effects. An iterative approach may be used to drive the system away from multiple line constraints as follows: at each step, the scanning algorithm identifies a desirable direction to move within the space of conventional generation output variables, and a step is taken in that direction. Iteration ceases when 1) it is no longer possible to step away from one thermal constraint without moving just as far towards another, 2) the required changes in conventional generation become too large, or 3) the goal of robustness to renewable fluctuation is achieved.[13]

MPC and related online adaptive methods tend to work around the two primary shortcomings of temporal deviation scanning. First, the algorithm is unable to ensure generators have non-negative output; second, it cannot guarantee that a reduction in one line's temperature will not increase the temperatures of other lines. In the first case, it is easy to observe whether generator outputs are negative; if they are, the pattern identified by the algorithm is clearly unrealistic. In the second case, it is straightforward to use our approximate line temperature model (or even the full IEEE 738 heat balance equation) to determine whether a re-dispatch suggested by our algorithm would cause any lines to overheat. In both cases, algorithm output will not be directly useful, but optimization terminates quickly and with certainty in any case. If a scenario results in negative generator outputs or causes overall system vulnerability to rise, it may simply be discarded, and iteration may continue in another direction.

3.5.5.4 Results

We tasked MPC with keeping all transmission lines in the 118-bus test network below $90°\mathrm{C}$ for 24 hours, in the presence of the demand and wind fluctuations described previ-

[13] Identification of a "desirable direction" at a given time step supposes that the algorithm can identify re-dispatch patterns that will cool multiple lines simultaneously. While this is often the case (as shown in our MPC case study), it is not guaranteed in general.

ously. Figure 3.10 illustrates the results. The top plot depicts temperature trajectories for all transmission lines in the absence of re-dispatch intervention (i.e. with MPC disabled). By 1:30pm, three lines have exceeded the acceptable temperature threshold. A fourth overheats later in the evening due to a sharp increase in wind generation (see Figure 3.9). The middle plot shows temperature trajectories with MPC enabled, while the bottom plot illustrates generator output adjustments (control actions) used by MPC to prevent transmission lines from overheating. The two transmission lines that would have overheated around 11:00am instead approach the $90°\mathrm{C}$ threshold gradually, and remain at that temperature through the afternoon and evening. The third line that crosses the threshold in the top plot is fortunately also cooled by mid-morning dispatch changes. The line that reaches $90°\mathrm{C}$ around hour 20, however, still exceeds the temperature limit with MPC enabled. This is likely due to the effects of previous dispatch changes, and highlights a fundamental limitation of the method: temporal deviation scanning can only attempt to cool one line at a time. As discussed in Section 3.5.5.3, implementing multiple deviation patterns at once may result in cancellation, thereby limiting the method's effectiveness. Even so, we see that the reasonably-sized dispatch adjustments derived from temporal deviation scanning are sufficient to prevent multiple lines from overheating. Computational cost is also minimal: all computation required for this case study was completed in several seconds on a laptop.

3.6 Conclusions

A computationally efficient approach for determining the smallest deviation from a wind power forecast that would drive a transmission line to its thermal limit over a specified time horizon was proposed. This concept is referred to as temporal deviation scanning. The problem setting incorporates models of the network, transmission line temperature dynamics, and wind forecast errors. Combining these models with a desire to keep deviations small results in a quadratically-constrained quadratic program (QCQP). The proposed analytic solution algorithm is suitable for realistic large-scale power systems.

Three case studies were presented to demonstrate the ability of temporal deviation scanning to address system challenges. First, the algorithm was used to identify the lines which are most vulnerable to renewable generation fluctuations, subject to correlations between wind-farms and time steps. Second, the algorithm was used to orchestrate conventional generators to reduce the temperature of a line operating above its thermal limit. These applications may be viewed as building blocks for higher-level methods. One such method, an MPC scheme, was implemented. Results show how temporal deviation scanning may

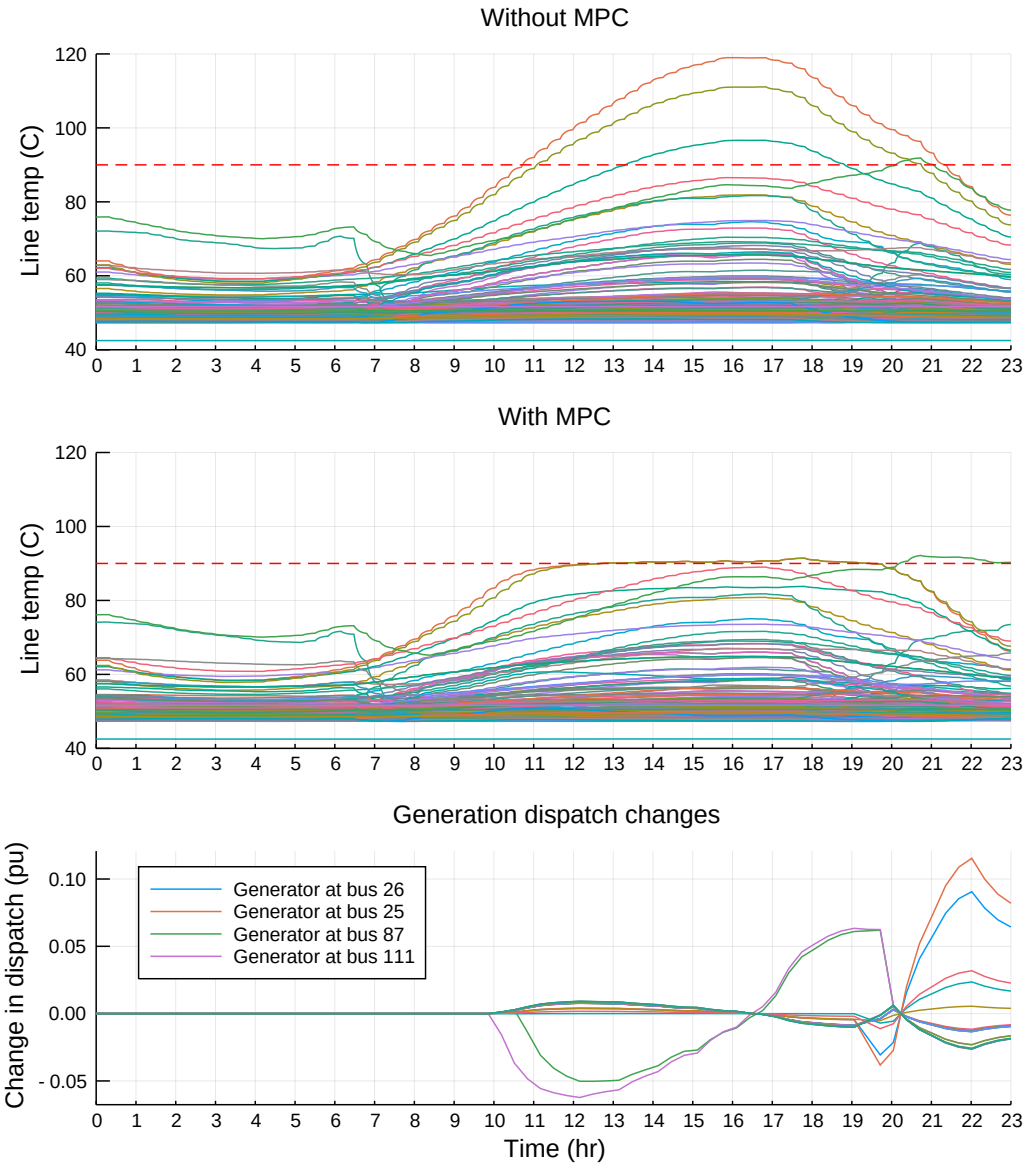


Figure 3.10: Normalized wind turbine power profiles from NREL's WIND dataset.

be used to manage line temperatures on a hot summer day by efficiently adjusting conventional generator outputs.

To ensure the proposed methods are adequate for addressing the concerns of a system operator like ERCOT, one important direction for future work involves application of the method to a larger, more realistic system. This would also involve the use of time series data (the subject of Chapter 4). Another branch of work follows in the direction of Section 3.5.5, developing higher-level iterative methods in which the deviation scanning algorithm is a sub-routine. Higher-level iterative methods are expected to efficiently produce feasible redispatch patterns that will enable system operators to reduce risk across large power

systems. The ability to alter generation output in a coordinated fashion to relieve a specific transmission line may make temporal deviation scanning useful for controlling distributed generation (DG). It may also be desirable to jointly address demand and wind uncertainty rather than modeling the latter alone, especially if extensive historical data are available. The flexibility of temporal deviation scanning suggests that there may be other fruitful applications we have not touched on.

Temporal deviation scanning incorporates a wide variety of time-varying system and thermal parameters. Depending on the application, the framework can quickly provide vulnerability information or assist an operator in responding to challenging conditions. These capabilities make it a powerful tool for system analysis and operation.

3.7 Appendix: Line Temperature Model

The change in temperature of an object is given by the heat balance equation, a differential equation that relates temperature change to the sum of various sources of heating. The IEEE 738 standard [45] provides such an equation for a transmission line,

$$\frac{dT_l}{dt} = \frac{1}{mC_p}\left[I^2R(T_l(t)) - q_c - q_r + q_s\right], \tag{3.28}$$

where:

- $T_l(t)$ (in °C) is the conductor average temperature,
- mC_p (in J/(m°C)) is the product of mass per unit length and specific heat,
- $I^2R(T_l(t))$ (in W/m) is the heat rate due to resistive heating,
- q_c (in W/m) is the convective heat loss, which is proportional to the temperature difference between the line and surrounding air,

$$q_c = \eta_c(T_l(t) - T_a), \tag{3.29}$$

- T_a (in °C) is the ambient temperature,
- q_r (in W/m) is the radiation heat loss, which is modeled by the fourth-order expression,

$$q_r = \eta_r\left[(T_l(t) + 273)^4 - (T_a + 273)^4\right], \tag{3.30}$$

- q_s (in W/m) is the solar heating. It is sufficient to assume q_s is fixed to a conservative constant corresponding to full, direct sun.

In building the QCQP line temperature model, the resistive heat rate term $I^2R(T_l(t))$ is replaced by f_{ij}^{loss}, the approximate line loss expression derived in [59]:

$$f_{ij}^{\text{loss}} \approx r_{ij}\left(\frac{\theta_{ij}}{x_{ij}}\right)^2 \times \frac{S_b}{3L_{ij}}, \tag{3.31}$$

where $\theta_{ij} = \theta_i - \theta_j$, $r_{ij} + \mathbf{j}x_{ij}$ is the per unit impedance of the line between nodes i and j, S_b is the system MVA base and L_{ij} is the length of line (i, j), in meters. Three DC power flow assumptions underpin (3.31): voltage magnitudes are all close to 1 pu, cosine may be approximated by its second-order Taylor expansion, and $x_{ij} \geq 4r_{ij}$. Thus, (3.31) approximates line losses using voltage angle differences.

Given an initial temperature $T_l(t_0)$, prediction of the conductor temperature $T_l(t_n)$ at a later time t_n is achieved by substituting (3.29)-(3.31) into (3.28) giving,

$$\frac{dT_l}{dt} = \frac{1}{mC_p}\big[f_{ij}^{\text{loss}} - \eta_c\left(T_l(t) - T_a\right) - \eta_r\left((T_l(t)+273)^4 - (T_a+273)^4\right) + q_s\big]. \tag{3.32}$$

This differential equation is still fourth-order in conductor temperature $T_l(t)$ due to the radiation term. Fortunately, q_r is approximately linear over the temperature range of interest (from ambient temperature to the conductor temperature limit), so it can be replaced by the conservative[14] linearization,

$$\tilde{q}_r = \eta_r\big[(T_m+273)^4 - (T_a+273)^4 + 4(T_m+273)^3(T_l(t) - T_m)\big], \tag{3.33}$$

where T_m is the average (midpoint) of the ambient and conductor limit temperatures in °C. After substitution of (3.33) into (3.32), the heat balance equation becomes linear in conductor temperature, having the form,

$$\frac{dT_l}{dt} = c_1 T_l(t) + c_2, \tag{3.34}$$

[14]Because a transmission line is hotter than surrounding air, radiation tends to decrease line temperature. Thus, a conservative approach will underestimate q_r. Plotting (3.33) shows that it does indeed underestimate q_r.

where c_1 and c_2 are defined as,

$$c_1 = \frac{1}{mC_p}\left[-\eta_c - 4\eta_r(T_m + 273)^3\right] \tag{3.35a}$$

$$c_2 = \frac{1}{mC_p}\Big[f_{ij}^{\text{loss}} + \eta_c T_a - \eta_r\big[(T_m + 273)^4 - (T_a + 273)^4 - 4T_m(T_m + 273)^3\big] + q_s\Big]. \tag{3.35b}$$

If line losses, ambient temperature and solar heat rate are constant, (3.34) has the solution:

$$T_l(t) = \Big(T_l(t_{init}) + \frac{c_2}{c_1}\Big)e^{c_1(t-t_{init})} - \frac{c_2}{c_1}, \tag{3.36}$$

for $t \geq t_{init}$. Note that c_2 is influenced by power flow (via f_{ij}^{loss} according to (3.31)), but c_1 is not.

The line temperature formulation presented in Section 3.1.2 is obtained by assuming $f_{ij}^{\text{loss}}(t_k)$ remains constant over the time interval $[t_k, t_{k+1})$. Given the initial temperature $T_l(t_0)$, (3.36) then gives $T_l(t_1)$, which can be used to give $T_l(t_2)$, and on to $T_l(t_n)$. Straightforward algebraic manipulation yields the form given by (3.1) and (3.2).

3.8 Appendix: Lagrangian Multiplier Proof

3.8.1 Statement

The following two optimization problems are equivalent:

$$\begin{aligned}
(P) \qquad & \min \quad o(\mathbf{w}) = \mathbf{w}^\top \hat{\mathbf{B}}\mathbf{w} - 2\hat{\mathbf{b}}^\top \mathbf{w} \\
& \text{s.t.} \quad \mathbf{w}^\top \mathbf{w} = c \\
(P') \qquad & \min \quad u \\
& \text{s.t.} \quad \mathbf{w}^\top \mathbf{w} = c && \text{(3.37a)} \\
& \qquad\quad \hat{\mathbf{B}}\mathbf{w} = u\mathbf{w} + \hat{\mathbf{b}}. && \text{(3.37b)}
\end{aligned}$$

Note that $\hat{\mathbf{B}}$ is diagonal, problem (P) is equivalent to (3.19), and (3.37) is equivalent to (3.21).

3.8.2 Proof

It is sufficient to prove that if $(u_1, \mathbf{w}_1)$ and $(u_2, \mathbf{w}_2)$ satisfy the first-order conditions (3.37b)-(3.37a) with $u_1 < u_2$, then

$$o(\mathbf{w}_1) = \mathbf{w}_1^\top \hat{\mathbf{B}} \mathbf{w}_1 - 2\hat{\mathbf{b}}^\top \mathbf{w}_1 < \mathbf{w}_2^\top \hat{\mathbf{B}} \mathbf{w}_2 - 2\hat{\mathbf{b}}^\top \mathbf{w}_2 = o(\mathbf{w}_2).$$

The proof uses a technique from [51]. From (3.37b),

$$\hat{\mathbf{B}} \mathbf{w}_1 = u_1 \mathbf{w}_1 + \hat{\mathbf{b}} \tag{3.38a}$$

$$\hat{\mathbf{B}} \mathbf{w}_2 = u_2 \mathbf{w}_2 + \hat{\mathbf{b}}. \tag{3.38b}$$

Pre-multiplying both (3.38a) and (3.38b) by $\mathbf{w}_1^\top$ and $\mathbf{w}_2^\top$ gives,

$$\mathbf{w}_1^\top \hat{\mathbf{B}} \mathbf{w}_1 = u_1 c + \mathbf{w}_1^\top \hat{\mathbf{b}} \tag{3.39a}$$

$$\mathbf{w}_2^\top \hat{\mathbf{B}} \mathbf{w}_2 = u_2 c + \mathbf{w}_2^\top \hat{\mathbf{b}} \tag{3.39b}$$

$$\mathbf{w}_1^\top \hat{\mathbf{B}} \mathbf{w}_2 = u_2 \mathbf{w}_1^\top \mathbf{w}_2 + \hat{\mathbf{b}}^\top \mathbf{w}_1 \tag{3.39c}$$

$$\mathbf{w}_2^\top \hat{\mathbf{B}} \mathbf{w}_1 = u_1 \mathbf{w}_2^\top \mathbf{w}_1 + \hat{\mathbf{b}}^\top \mathbf{w}_2, \tag{3.39d}$$

where (3.37a) was used to replace instances of $\mathbf{w}_1^\top \mathbf{w}_1$ and $\mathbf{w}_2^\top \mathbf{w}_2$ with c. Now subtract (3.39b) from (3.39a) to obtain,

$$(\mathbf{w}_1^\top \hat{\mathbf{B}} \mathbf{w}_1 - \mathbf{w}_1^\top \hat{\mathbf{b}}) - (\mathbf{w}_2^\top \hat{\mathbf{B}} \mathbf{w}_2 - \mathbf{w}_2^\top \hat{\mathbf{b}}) = (u_1 - u_2)c.$$

Add $\hat{\mathbf{b}}^\top \mathbf{w}_2$ and subtract $\hat{\mathbf{b}}^\top \mathbf{w}_1$ to yield,

$$o(\mathbf{w}_1) - o(\mathbf{w}_2) = (u_1 - u_2)c - \mathbf{w}_1^\top \hat{\mathbf{b}} + \mathbf{w}_2^\top \hat{\mathbf{b}},$$

then substitute (3.39c) and (3.39d),

$$\begin{aligned} o(\mathbf{w}_1) - o(\mathbf{w}_2) &= (u_1 - u_2)c + (u_2 - u_1)\mathbf{w}_1^\top \mathbf{w}_2 \\ &= (u_1 - u_2)(c - \mathbf{w}_1^\top \mathbf{w}_2). \end{aligned}$$

Note that,

$$\|\mathbf{w}_1 - \mathbf{w}_2\|^2 = \mathbf{w}_1^\top \mathbf{w}_1 + \mathbf{w}_2^\top \mathbf{w}_2 - 2\mathbf{w}_1^\top \mathbf{w}_2 = 2c - 2\mathbf{w}_1^\top \mathbf{w}_2,$$

so,

$$o(\mathbf{w}_1) - o(\mathbf{w}_2) = \frac{1}{2}(u_1 - u_2)\|\mathbf{w}_1 - \mathbf{w}_2\|^2.$$

Because $u_1 < u_2$, the right-hand side is negative. The objective value $o(\mathbf{w}_1)$ is therefore less than that of $\mathbf{w}_2$. It may be concluded that (P) and (P') are equivalent.

CHAPTER 4

Time series

4.1 Introduction

Temporal analysis techniques such as the temporal deviation scanning algorithm rely on time series data to track changes in demand and generation. While previous chapters focused on the challenges associated with fluctuating system conditions, this chapter is concerned with properties of the time series that model such changes. Electric load profiles, for example, possess characteristics that vary with time resolution and level of aggregation. It should therefore be possible to detect unrealistic time series datasets, or even individual load profiles. In this chapter, data science techniques are applied to realistic and synthetic electric load time series in order to distinguish them.

Section 4.2 describes the five electric load profile data sources we will use. Section 4.7 details the process of converting each dataset to a single format in preparation for analysis. Section 4.3 illustrates typical load profile characteristics related to human behavior and climate. Section 4.4 presents consistency metrics based on patterns across a dataset, while Section 4.5 describes metrics for individual time series. Section 4.6 concludes.

4.2 Data sources

There is a distinction between *original load profiles*, obtained directly from measurement, and *derived time series*, which are derived from original load profiles. The latter are often found in synthetic or disaggregated load profile datasets. While our five datasets contain roughly 200 time series, only around 40 are original load profiles.

4.2.1 WECC 240 (aggregated and disaggregated)

The aggregated WECC 240 dataset consists of 21 original load profiles from CAISO, aggregated at the area level. In addition to the aggregated WECC dataset, a disaggregated WECC dataset attempts to model the load at each of the 132 load buses. The disaggregated dataset was derived by assigning each bus in the WECC 240 network a scaled copy of the load profile for the area to which it belongs (21 areas total). Scaling was done according to load participation factors. As we will show, analysis of the disaggregated dataset can detect that disaggregation was performed, and accurately estimate the number of original load profiles from which the larger set of derived time series was obtained. All WECC electric load time series begin January 1, 2004 at 1 AM; and end January 1, 2005 at midnight. Time resolution is one hour, so there are 8,784 samples per time series (as 2004 was a leap year with 366 days).

4.2.2 RTS 96 (case73)

The RTS 96 load profile dataset is unrealistic. As subsequent analysis will show, each of its time series was generated by a single function, not obtained from independent measurement (or indeed any measurement at all. The 51 RTS 96 electric load time series are at a one-hour resolution. They begin January 1, 1996 at midnight, and end December 29, 1996 at 11 PM (December 30 and 31 are missing). This results in a total of 8,736 samples per time series.

4.2.3 ERCOT

The Energy Reliability Council Of Texas (ERCOT) provides public access to load data online. We downloaded the last sixteen annual load data files and concatenated them to obtain a single dataset with a time index spanning 2002-2018. Each of the eight load profiles corresponds to one of ERCOT's weather zones. Together, these zones span most of Texas. The exact time range for each column is 1 AM January 1, 2002 to 11 PM April 30, 2018; as with previously mentioned datasets, the sample rate is one per hour.

4.2.4 RTE GRID DATA partnership

These twelve load profiles were obtained via measurement by RTE in France. Each time series corresponds to one of the country's administrative divisions. The original files are from RTE. The time range extends from midnight on January 1, 2013 to 11:30 PM on July

31, 2017; the time resolution is 30 minutes. This results in a total of 80,304 samples per time series.

4.2.5 RTS 96 GMLC

These load profiles were synthesized, and are associated with a modified version of the RTS 96 network. There is a high degree of aggregation: the three columns correspond to the three identical areas of the network. This dataset is noteworthy for its high time resolution of five minutes. The time range begins at midnight on January 1, 2020, and ends at 11:55 PM on December 31, 2020. This results in 105,408 samples per time series.

4.2.6 Supplementary data for RTE analysis

The RTE dataset is especially valuable for its breadth and realism. To take full advantage of this data, we augmented it with population and temperature data from other sources. Texas A&M University maintains an agricultural database that contains temperature data from around the world [62, 63]. We used the online interface they provide to identify a weather station near the center of each of RTE's twelve load aggregation regions, then downloaded daily temperature data for 2016-2018. Population data for these same administrative regions may be obtained from France's National Institute of Statistics and Economic Studies (INSEE). We downloaded 2015 estimated population data from a website where the desired INSEE data was available as a table.[1]

4.2.7 Other data details

There are several other details associated with importing and using the data. Some metadata are missing. There is no information on the size of each load, for example. Units are also missing, so it is unclear whether a column contains kilowatt or megawatt data (though MW can be reasonably assumed). Another issue relates to different time resolutions. The WECC and RTS 96 datasets have a one-hour resolution, RTE has a 30-minute resolution, and RTS 96 GMLC has a five-minute resolution. In comparative analysis of the four datasets, we down-sampled the latter two datasets to one-hour resolution to match WECC and RTS 96. To down-sample a dataset, we simply threw away samples between hourly timestamps. Using these in-between samples to perform smoothing would alter our analysis, especially in the frequency domain. Each dataset also has a different time horizon. The WECC and RTS 96 GMLC datasets span 366 days (each corresponds to a leap year), while RTS 96

[1] `https://www.citypopulation.de/France-Cities.html`

data spans 364 days (also a leap year, but missing December 30 and 31). RTE data spans 1,673 days, or roughly 4.5 years. Where computationally necessary or visually appropriate, we truncated all datasets to 8,736 hours, the time span of the limiting RTS 96 dataset.

4.3 Data exploration

4.3.1 Seasonal variation

The most important feature of load profile data is its connection to human behavior. This manifests in *seasonal variation*, a catch-all term in time series analysis for fixed-period cyclical variation.[2] Daily variation is strongest, with most load profiles also exhibiting weekly variation from weekdays to weekends, and variation from winter to summer. Roughly speaking, daily variation takes the form of higher load during the day and lower load at night. From weekday to weekend, many load profiles lose the early-morning peak associated with people making coffee and preparing for work. Seasonal variation depends on the region. In warm weather locales, load profiles peak during summer afternoons as air conditioners are fully utilized. In colder areas, daily load profiles are more steady, and peaks are associated with heating rather than cooling.

Figure 4.1 illustrates the division of load profiles into four fairly distinct classes. The two factors with the greatest influence on load profile shape are season (summer vs winter) and day of the week (weekday vs weekend). The WECC network is located on the west coast of the United States, where the dominant HVAC need is for air conditioning in the summer. The load profile of a typical summer day in the WECC network is roughly sinusoidal, with a peak occurring around 4:30 PM. Electricity usage drops considerably in the winter. Between weekdays and weekends, the most significant difference has to do with the pre-work peak around 7:30 AM. The ubiquitous coffee habit likely plays a role. Note that "summer" and "winter" are connected more to weather than the calendar, while Figure 4.1 was generated by designating summer to be May-October and winter to be November-April. This explains the presence of summer-like curves in the lower two sub-figures. Generating a plot like 4.1 for the RTS 96 GMLC dataset shows that it is similar to the WECC dataset. The ERCOT load profile shown in Figure 4.4 is also similar to WECC data, as Texas is also a summer-peaking region.

RTE load profile data comes from France, where cold winters make heating a necessity. Figure 4.2 illustrates the effect this has on load profiles. The load never exceeds 70% of

[2]Other forms of variation include the *trend*, a broad shift across a time series; and *cyclical variation*, which is periodic but may not have a fixed period or as consistent an effect as seasonal variation.

its peak value in the summer months, but is above 70% *on average* in the winter. The data also exhibit weekday morning peaks.

Load profiles exhibit seasonal variation in most climates. This makes the RTS 96 data in Figure 4.3 suspicious. It is apparent that there are only a few unique daily load shapes in this yearlong load profile; each day is just a scaled copy of a prototypical shape. In other words, the dataset is low-rank. This fact is revisited in Sections 4.4.1 and 4.4.2.

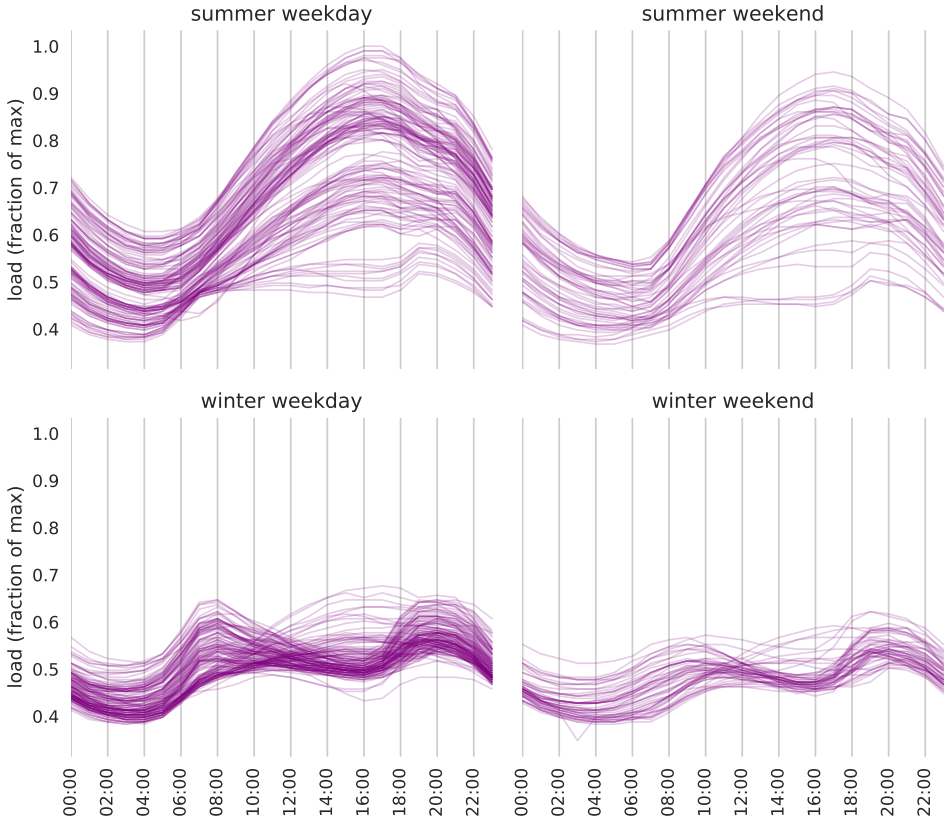


Figure 4.1: Breakdown of one yearlong WECC load profile (United States west coast, warm climate) by season and weekday.

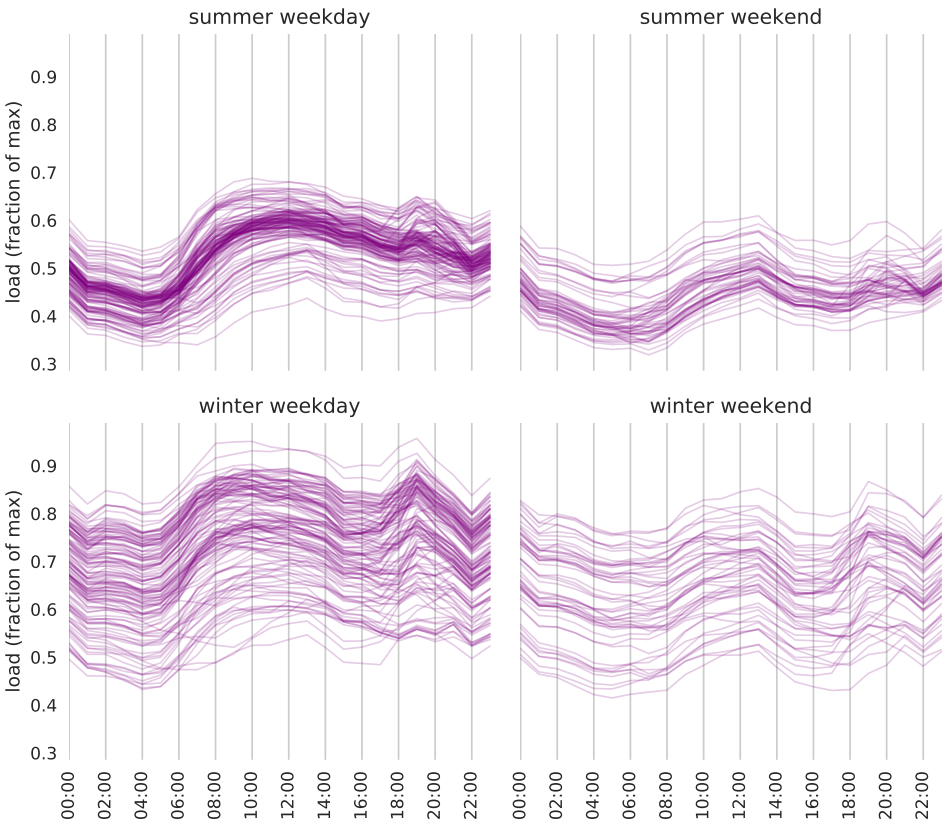


Figure 4.2: Breakdown of one yearlong RTE load profile (France, colder climate) by season and weekday.

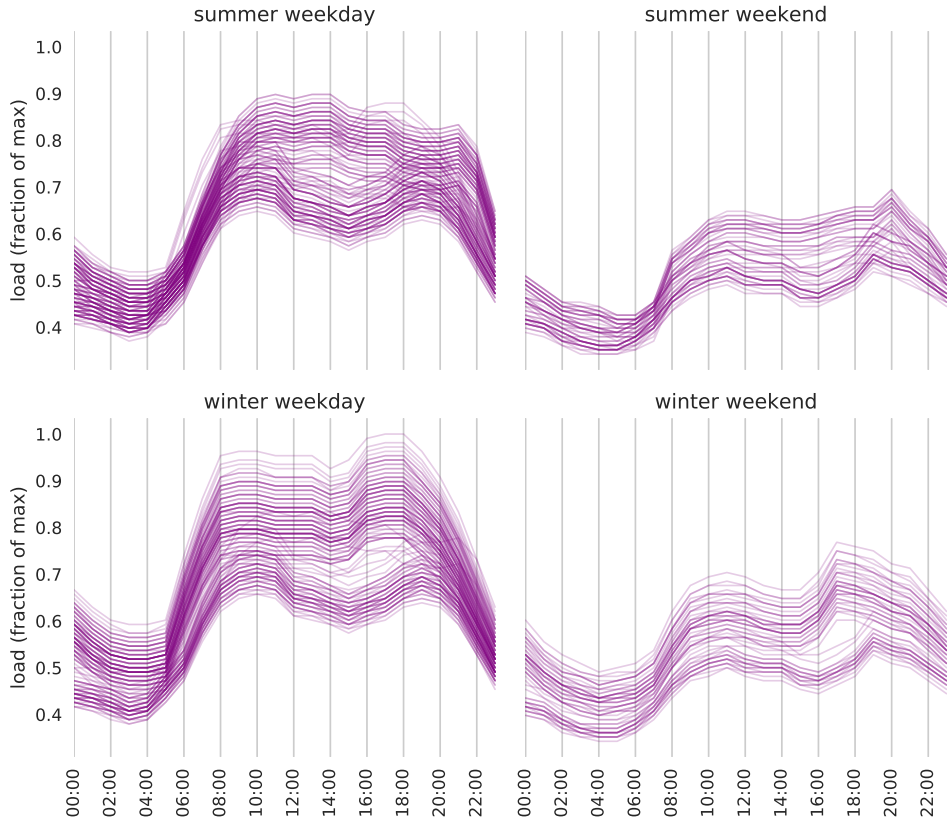


Figure 4.3: Breakdown of one yearlong RTS 96 load profile by season and weekday.

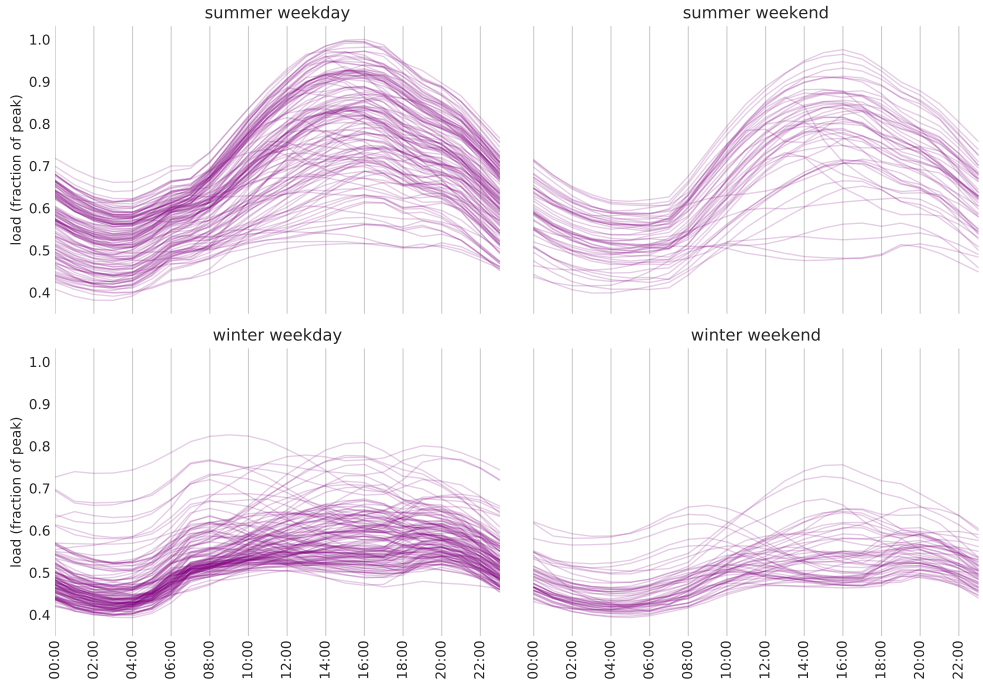


Figure 4.4: Breakdown of one year of a load profile from the ERCOT dataset.

4.3.2 Temperature and population

Electric load is closely tied to outdoor temperature and population. Figure 4.5 illustrates these relationships. Each sub-figure is an overlay of all daily load profiles for a particular administrative division of France during 2016. Sub-figures are labeled with, and sorted by, 2015 population estimates of the twelve divisions. All sub-figures share the same axis limits. Individual daily load profiles are colored according to the average temperature for the day and region (according to the agricultural database maintained by Texas A&M University; see Section 4.2.6). As a winter-peaking region, France uses the most electricity on the coldest days, and the least electricity on room-temperature days. As population decreases, load profiles appear flatter provided the vertical scale is unchanged. Figure 4.6, a normalized version of Figure 4.5, shows that the appearance of flatness in the last few subplots of Figure 4.5 is a result of axis scaling. Figure 4.7 focuses only the most- and least-populated divisions, and provides a closer look at seasonal variation in Paris (Ile-de-France).

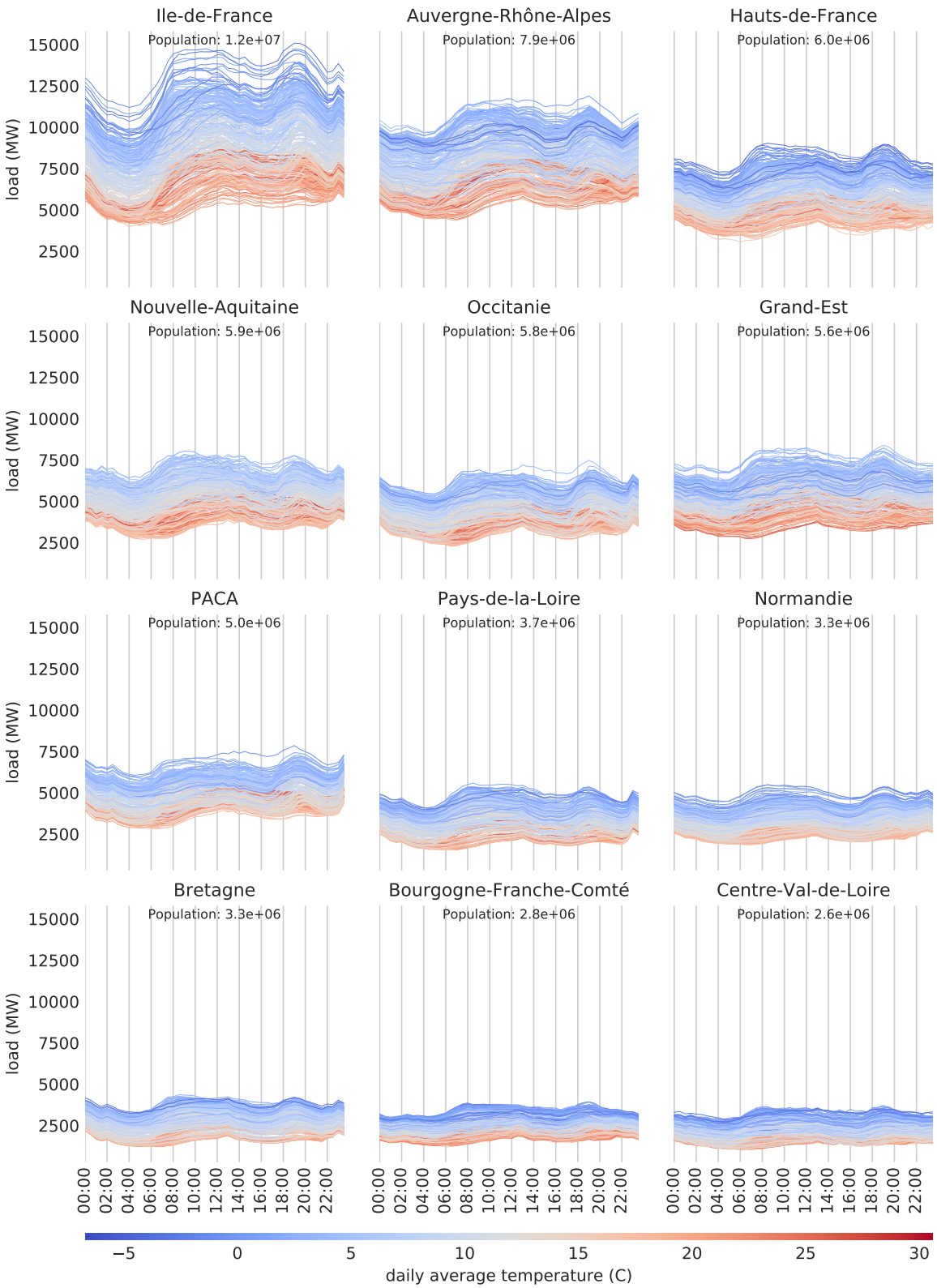


Figure 4.5: 2016 load profiles for twelve administrative regions of France. Regions are sorted (left-to-right and top-to-bottom) by 2015 estimated population. Individual daily load profiles are colored according to temperature.

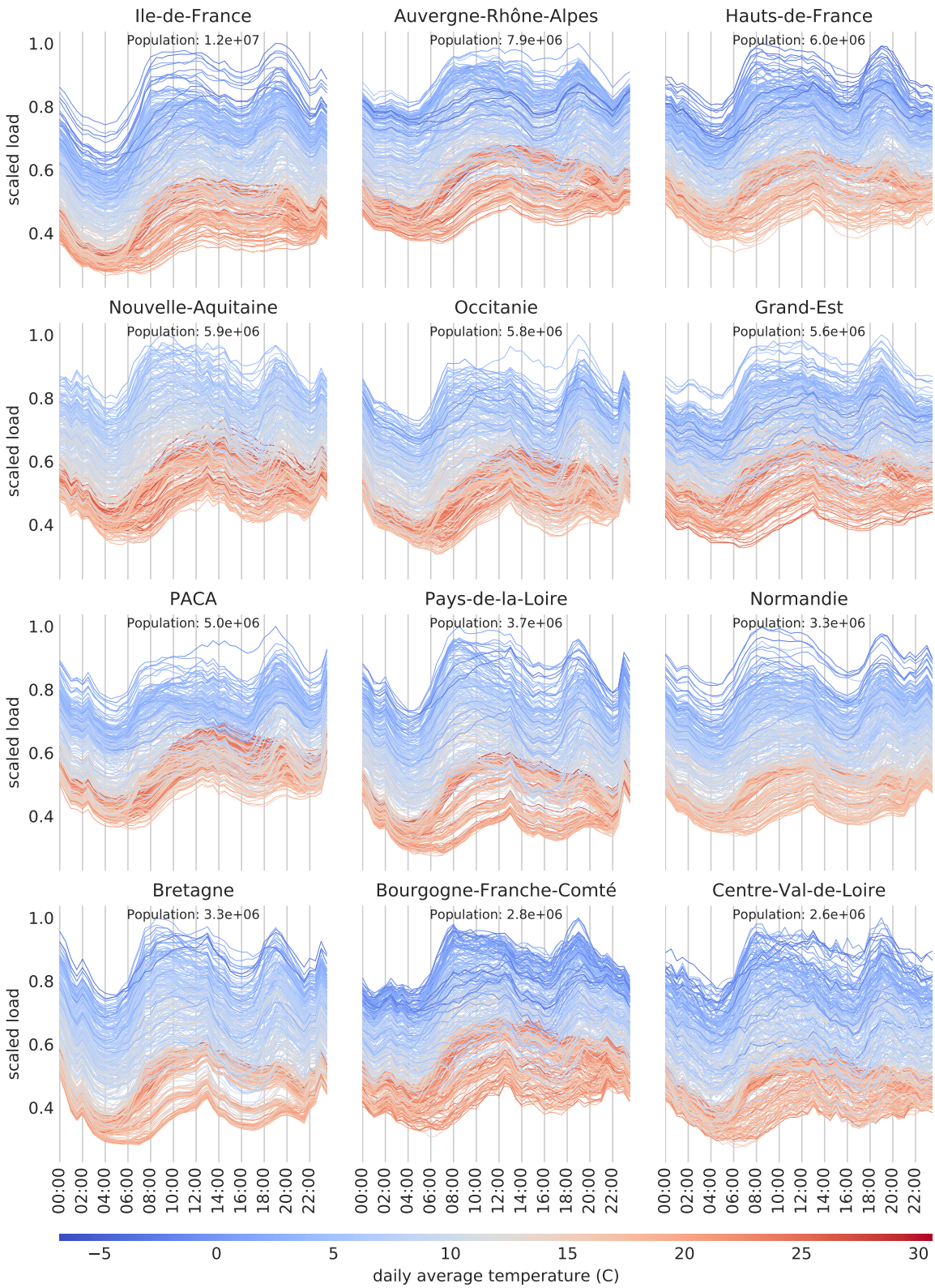


Figure 4.6: Normalized version of 4.5. Each division's load profile was scaled by its maximum value; each scaled profile takes values in $[0, 1]$.

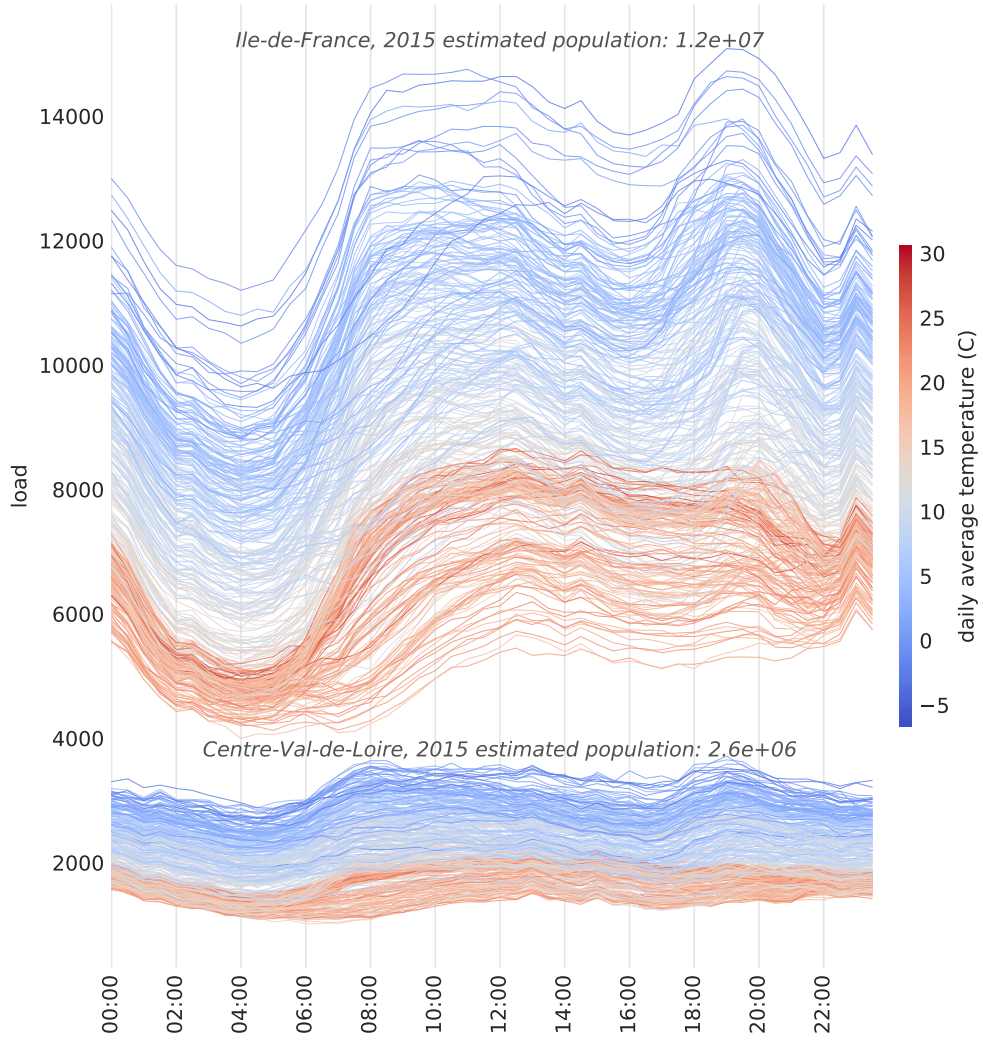


Figure 4.7: 2015 load profile and temperature data for France's most- and least-populated divisions.

4.4 Dataset consistency

Searching for patterns across a load profile dataset can reveal artificiality. Original load profiles may be correlated with each other, but only derived load profiles will exhibit linear dependence. In this section we propose two metrics for detecting derived time series in load profile datasets.

4.4.1 Cross-correlation analysis

Time domain tests effectively reveal correlation within a dataset. While some correlation is expected in realistic load data, excessive correlation suggests that multiple locations share one time series shape (an unrealistic feature of some load profile datasets). For a time series DataFrame with load profile data, the correlation matrix denotes the matrix whose (i, j)-th element is the Pearson correlation coefficient between columns i and j.

Consider the correlation matrix of the WECC load profile dataset in Figure 4.8. The triangular patches of dark elements (at or near 1.0 in value) along the diagonal suggest rank deficiency. This manifests throughout the rest of the matrix in a plaid pattern. It appears that some load profiles are scaled versions of others. In fact, the full WECC load profile dataset was generated from 21 aggregated load profiles, whose correlation matrix is shown in Figure 4.9.

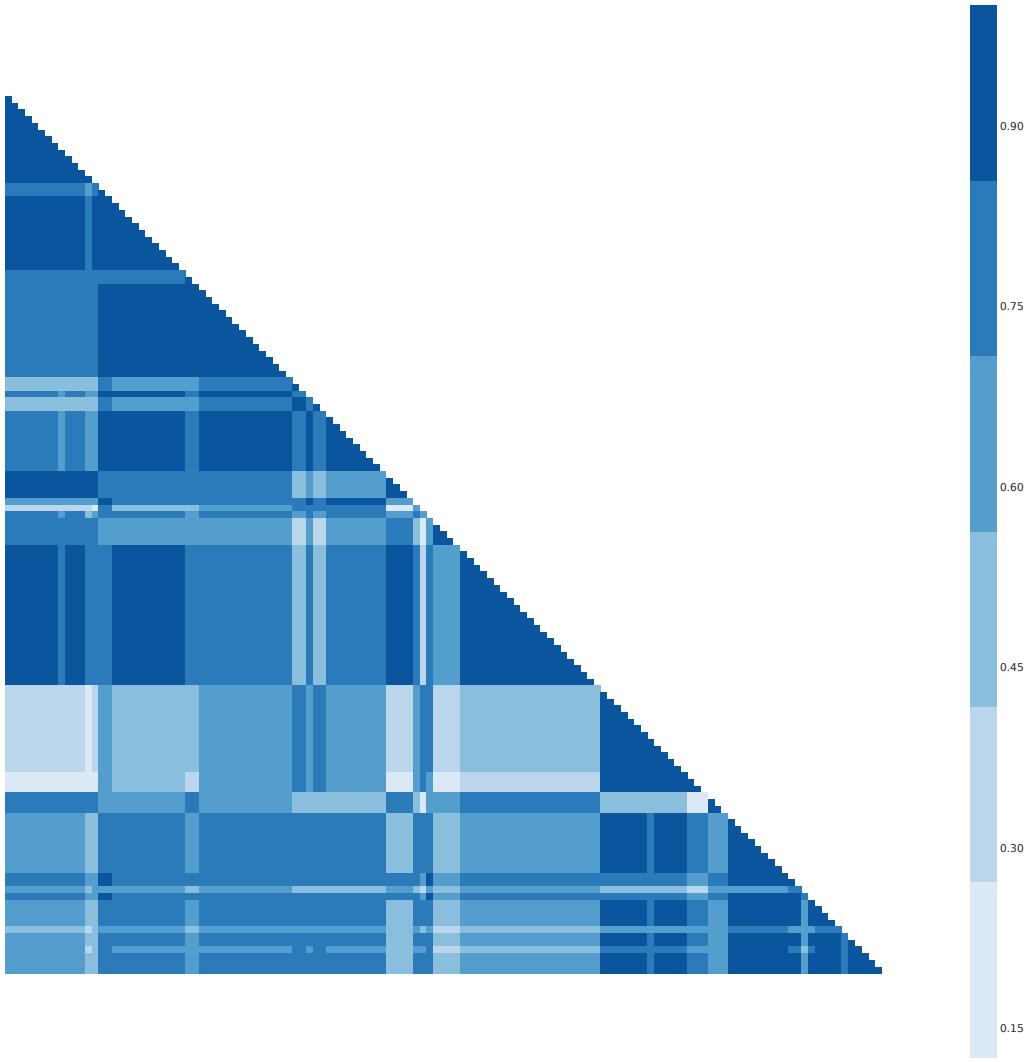


Figure 4.8: WECC dataset correlation matrix.

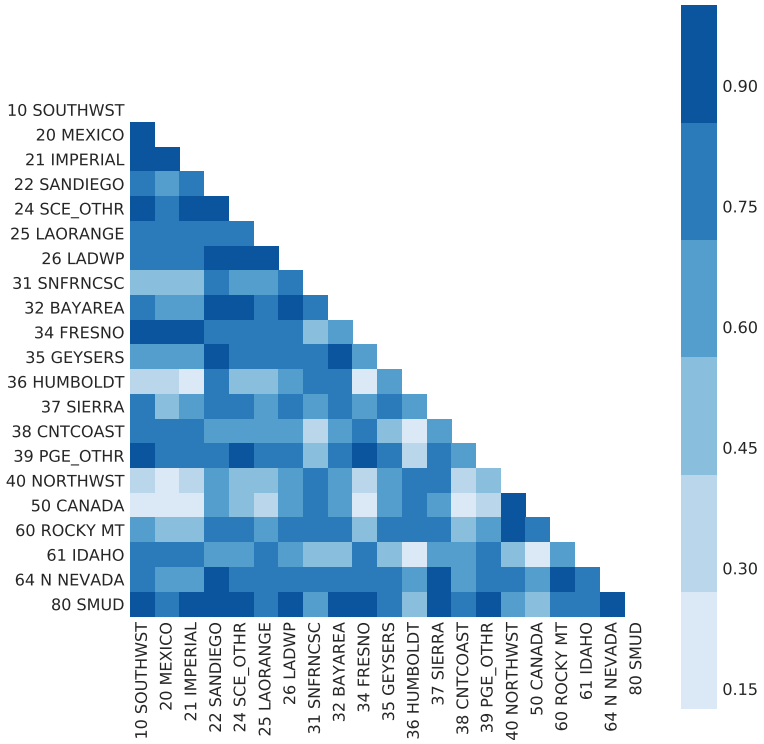


Figure 4.9: Aggregated WECC dataset correlation matrix.

Figure 4.10 shows that the RTS 96 load profile dataset has low rank as well. Note the color scale: all elements are greater than 0.9992, which suggests that all columns are (possibly scaled) copies of one time series. Figure 4.10 also reveals a tiling pattern. One can construct the full matrix by tiling the sub-matrix boxed in red. This suggests that the dataset was synthesized by copying the load profiles for one area to the other two areas of the RTS 96 network.

The correlation matrix corresponding to RTE load profiles is shown in Figure 4.11. There is no discernible pattern in the data that might suggest linear dependence between any columns. The strong correlation between areas is surprising at first glance, but makes sense given the relative flatness of load profiles in winter-peaking regions. The correlation matrix of the ERCOT dataset, pictured in Figure 4.12, is similarly full-rank. The correlation matrices for the RTE and ERCOT datasets have a "realistic look" when compared to the patchy correlation matrices for the synthetic datasets (WECC and RTS 96). This intuition will be formalized in Chapter 5.

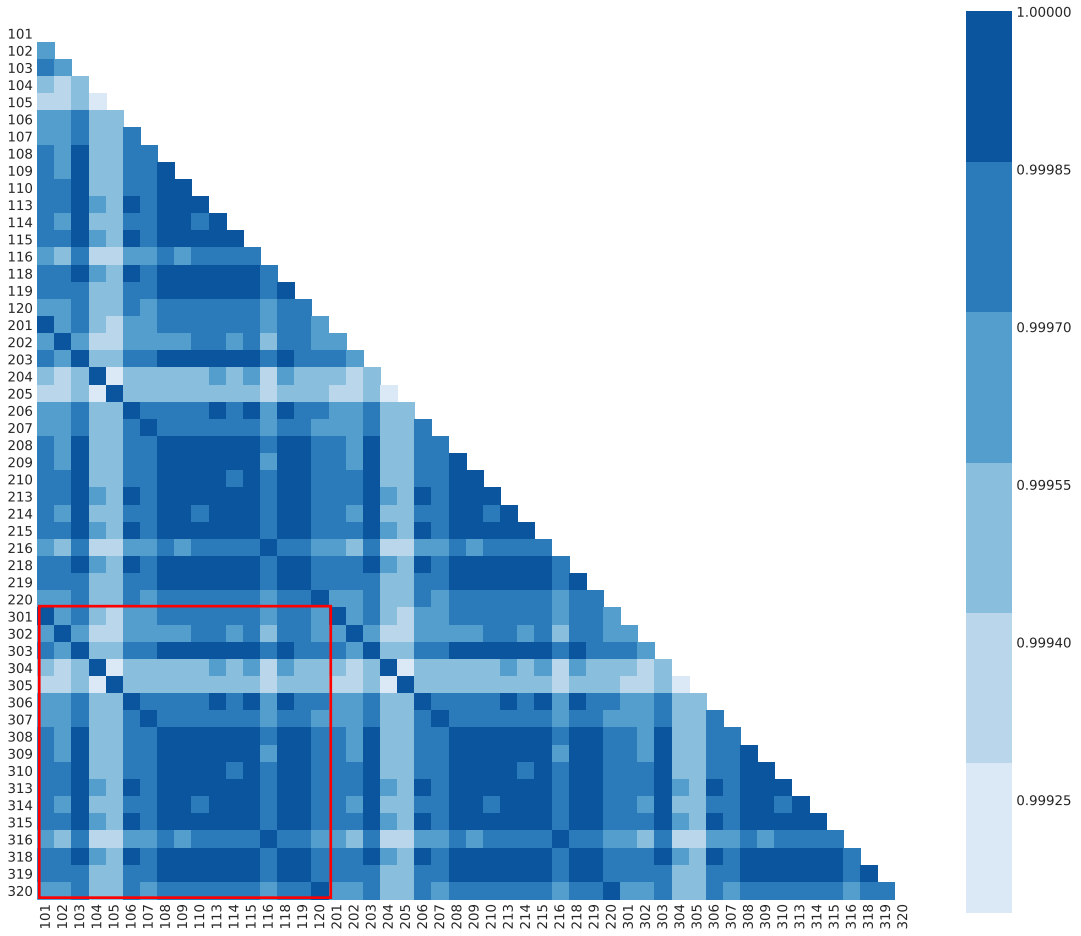


Figure 4.10: RTS 96 load profile correlation matrix. All elements are effectively equal to 1; the narrow color scale highlights tiny differences between elements. Note the sub-matrix boxed in red, which may be tiled to obtain the full matrix.

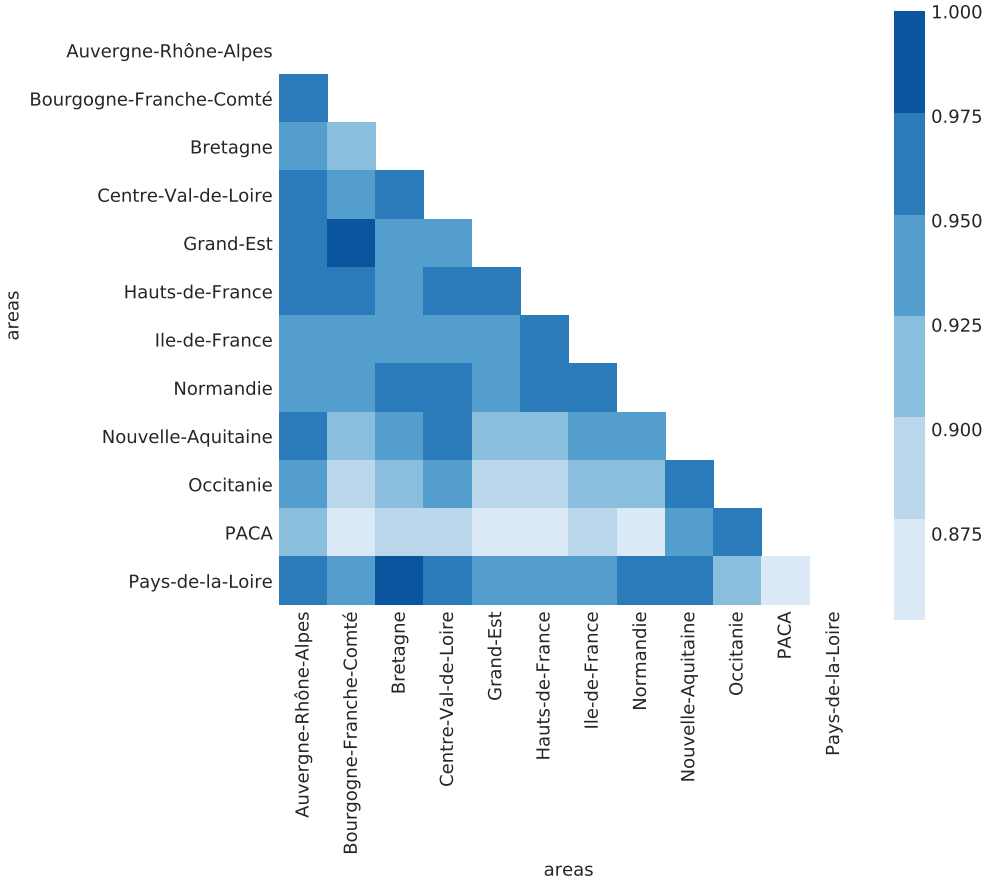


Figure 4.11: RTE dataset correlation matrix.

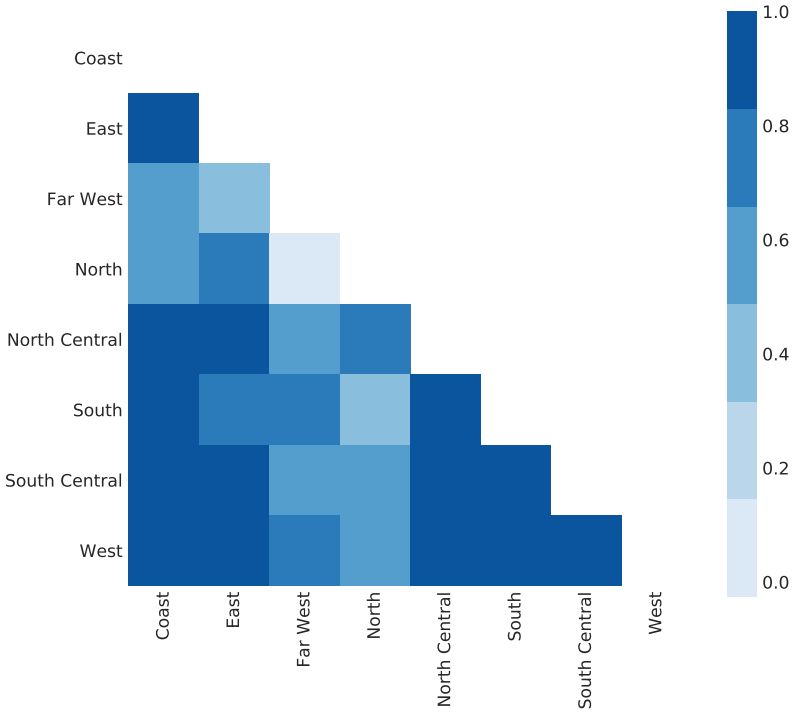


Figure 4.12: ERCOT dataset correlation matrix.

4.4.2 Singular values

The singular value decomposition also reveals correlation between buses and areas. If all columns of a load profile dataset are based on a single prototype load profile, then the rank of the data matrix will be close to 1. If the columns are linearly independent, then the matrix will have full rank (equal to the number of columns). Figure 4.13 illustrates singular values for the six datasets. The WECC dataset is rank-deficient by 4, which suggests that four of the time series are linearly dependent on others. The less-realistic RTS 96 data stands out in this test, with only 16 linearly-independent columns. This reinforces the earlier finding that all columns of this dataset are correlated to an unrealistic degree. The remaining two datasets, RTE and RTS 96 GMLC, have full rank. Note that only the first year of RTE data was analyzed, as the SVD becomes too computationally expensive for multi-year data.[3]

It is noteworthy that the RTS 96 appears to have rank 17 in Figure 4.13, even though it is a rank-1 dataset. Similarly, the triangular patches on the diagonal in Figure 4.8 suggest that the WECC dataset is low-rank, but Figure 4.13 indicates it is only rank-deficient by four. Both WECC and RTS 96 have a number of singular values on the order of $10^2 - 10^5$, much larger than 10^{-10} (understood to be zero) but also orders of magnitude smaller than the largest singular values. The WECC dataset has 20 singular values greater than 100, while the RTS 96 dataset has just one. The cause of these apparent discrepancies is quantization noise, which will be analyzed in detail in Chapter 5.

[3]With an efficient algorithm, the SVD of a $m \times n$ matrix requires $O(km^2 + k'n^3)$, where k and k' are constants. One can obtain the singular values alone with $O(mn^2)$ operations.

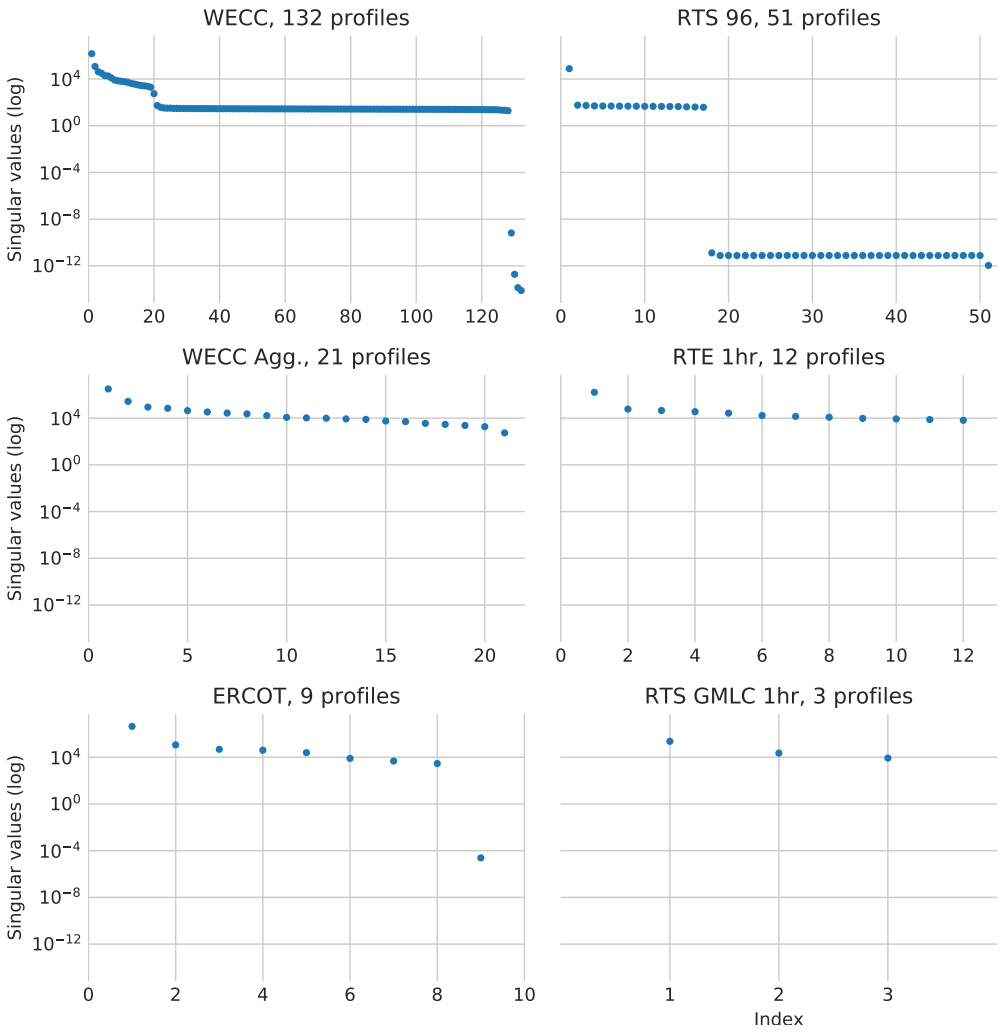


Figure 4.13: Singular values of each dataset, treating each time series a one column of a matrix.

4.5 Time series analysis

The metrics derived in Section 4.4 are based on patterns across a dataset. In this section we investigate individual time series characteristics using a few well-known types of analysis. Fourier analysis reveals periodicity, but fails to capture time-varying characteristics. The spectrogram simultaneously illustrates variation in time and frequency, but is difficult to distill into metrics. Singular Spectrum Analysis (SSA) effectively separates trend and noise components, and is more conducive to concise metrics. Finally, Seasonal Autoregressive Integrated Moving Average (SARIMA) modeling can reduce typical load profile characteristics down to a few parameters.

4.5.1 Fourier transform

The Fourier transform yields a frequency domain representation of time domain data. Before comparing frequency spectra, we ensured comparability across datasets by fixing time resolutions to one hour, and time spans to 8,736 hours (the time span of the shortest dataset). We then normalized each time series so its largest value is 1. Consider the five frequency spectra shown in Figure 4.14. Each trace represents the DFT of the first column of one of the datasets. The spectra reveal a few intuitive features of load profiles. All five traces have peaks at the once-per-day frequency. The once-per-week frequency component is more pronounced for some time series than others, and is accompanied by its own harmonics. Note the harmonics of the once-per-week fundamental frequency for the middle two spectra.

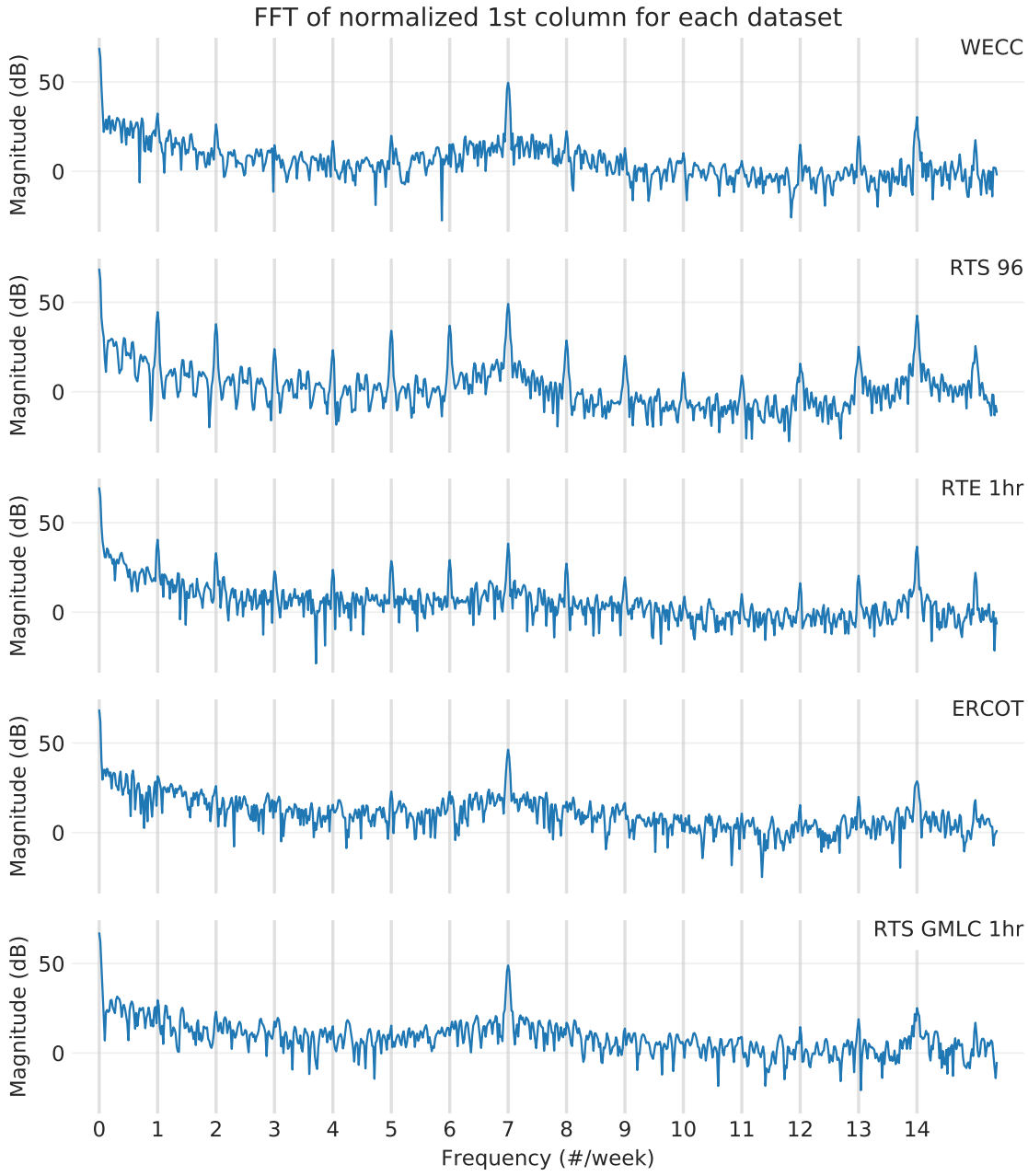


Figure 4.14: DFT coefficients for first column of each dataset, spanning the frequency range from 0 to twice per day. Columns were normalized such that their largest elements were all equal to 1.0.

Metrics derived from Fourier analysis are unlikely to be useful, as they fail to capture time-varying (e.g. seasonal) behavior. The spectrogram, which we will consider next, addresses this shortcoming via a trade-off between time and frequency resolution.

4.5.2 The spectrogram

The spectrogram is a function that assigns an intensity value to a given time and frequency. A spectrogram is naturally visualized as a two-dimensional heatmap. Due to the fundamental trade-off between time and frequency resolution, one must choose an appropriate time window before generating a spectrogram. Having experimented to identify appropriate parameters, we generated spectrograms for ERCOT and RTE load profile data. These are shown in Figures 4.15 and 4.16. Note in Figure 4.15 that the once-per-day frequency component is stronger in summer, when peak daily temperatures require substantial electric cooling. By simultaneously showing time and frequency variation, the spectrogram provides more insight than Fourier analysis. On the other hand, it is unclear how to condense spectrogram information into metric-style summaries of load profiles. In the next section we describe a means of decomposing a time series into trend and noise components, thereby moving towards a concise metric.

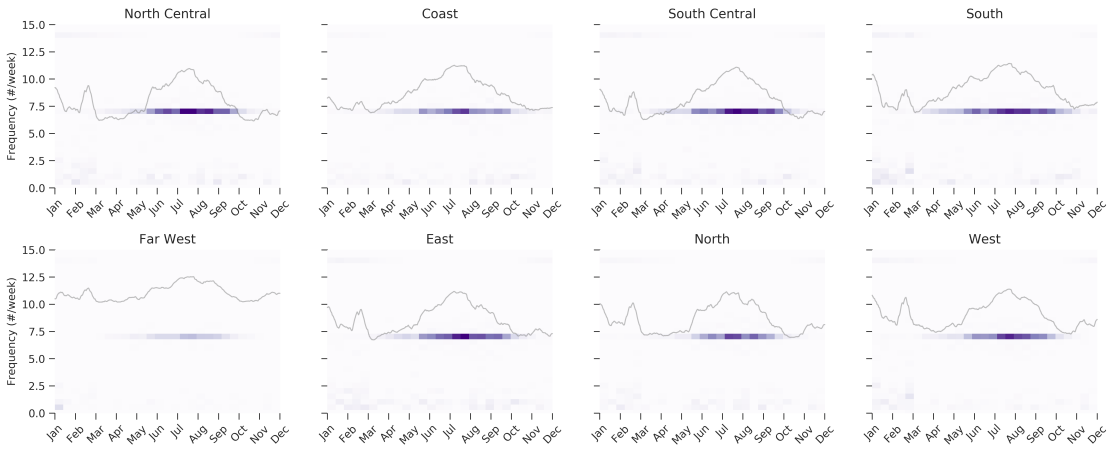


Figure 4.15: Spectrogram of one year of ERCOT load profile data. Gray traces are seven-day moving averages of time domain data. Note the seasonal variation of daily peak prominence.

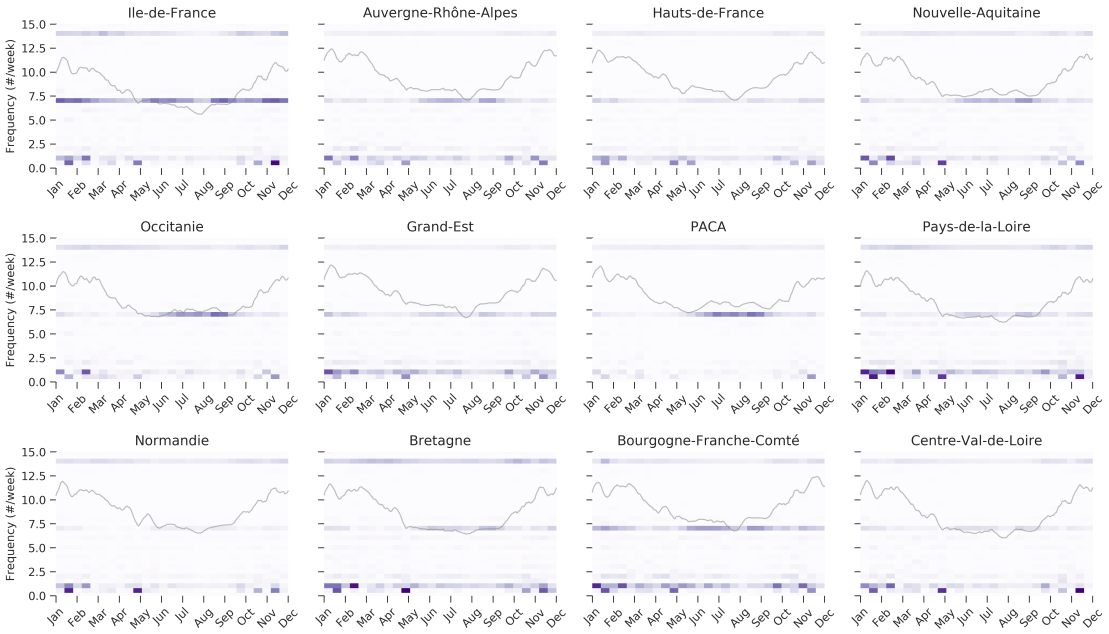


Figure 4.16: Spectrogram of one year of RTE load profile data. Gray traces are seven-day moving averages.

4.5.3 Singular spectrum analysis (SSA)

SSA is a model-free, non-parametric signal decomposition based on principal component ideas [64]. Rather than the sinusoidal basis signals of Fourier analysis, SSA provides "adaptive spectral filters associated with dominant oscillations" [65]. SSA is suitable for nonlinear and non-stationary signals, and has been applied to electric load data previously [66, 67, 68, 69].

Given a centered, length-N time series, SSA proceeds as follows. First, a window length (or "embedding dimension") $M < N/2$ is chosen. Next, the input signal is embedded in a "trajectory matrix" of lagged vectors of length M; each length-M vector is a lagged version of the previous one. The SVD is applied to this trajectory matrix to obtain eigen-triples (U_i, s_i, V_i), then eigen-gaps between consecutive singular values s_i are used to group these eigen-triples together (with the number of groups depending on how many SSA components one wishes to obtain). Finally, the embedding process used to generate the trajectory matrix is reversed via diagonal averaging to yield SSA components. Provided that an appropriate window length was chosen and the eigen-gaps were large enough, SSA will decompose the input signal into trend, oscillatory, and noise components.

It is important to choose an appropriate embedding dimension M. A higher value of M will capture oscillations with greater periods, and will better isolate noise; lower M makes eigen-grouping easier and analysis faster [65]. To quickly analyze hundreds of load profiles, we let $M = 24$ for all SSA analysis. Figure 4.17 helps to explain this choice. Note the clear eigen-gaps separating the eigen-groups $\{1\}$, $\{2-5\}$, $\{6-15\}$, and $\{16-24\}$ for the $M = 24$ case (in orange). When applying SSA with $M = 24$ to other load profiles, we observed similar eigen-gaps, which enabled us to use the same eigen-grouping across all load profiles. It makes sense that an embedding dimension of 24 is effective, as this is the period of daily oscillation (24 hours).

Figure 4.18 illustrates SSA components for an ERCOT load profile. The four components (which may be summed to obtain the original signal) are designated as trend (or large-period oscillations), 1/day, 2/day, and "harmonics + noise." This signal representation clearly shows the relative importance of each component, as well as the months during which it is strongest. Note that the seasonal component does not appear to be periodic, in contrast to the oscillatory appearance of the 1/day and 2/day components. The trend component at the top of Figure 4.18 indicates that ERCOT is a summer-peaking region, while the 1/day component confirms that May-October behavior is dominated by a once-per-day oscillation. In the colder months, the twice-per-day component is stronger. This corresponds to the daily load profiles illustrated in the lower-left subplot of Figure 4.4. Ideally, the last component ("harmonics + noise") should be small and stationary (exhibiting no

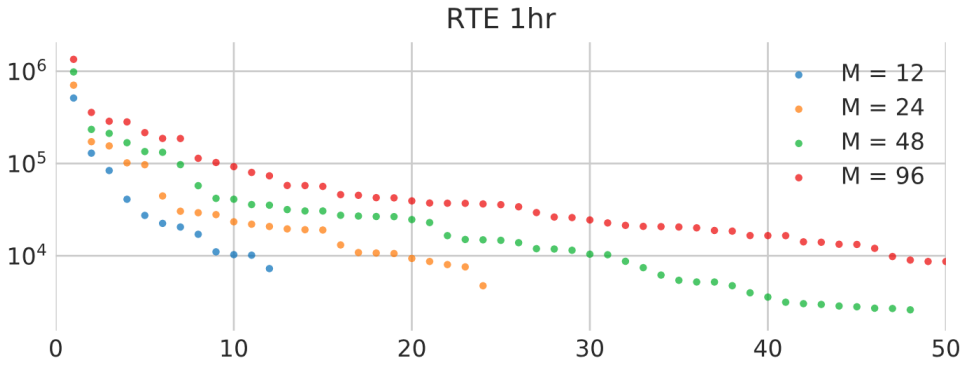


Figure 4.17: Eigen-gaps for various choices of embedding dimension M. The load profile being embedded is from the RTE dataset. Eigenvalues are on the vertical axis, with SSA component index on the horizontal axis.

time variation).

4.5.4 SARIMA modeling

Autoregressive Integrated Moving Average (ARIMA) modeling is a common time series technique, with some prior application to power systems [67]. The acronym stands for Auto Regressive (*AR*, regression of a signal against lagged versions of itself) Integrated (*I*, differencing to attain stationarity) Moving Average (*MA*, error is computed as a moving average). Seasonal ARIMA, or SARIMA, is the product of a regular ARIMA model with a second ARIMA model that captures seasonal variation; this extends ARIMA modeling to accommodate seasonal variation with a known period. SARIMA modeling involves two steps: first, one must choose *AR*, *I*, and *MA* orders for the base and seasonal ARIMA models. A high enough *I* order is selected to ensure that the differenced time series is stationary. Then the AR and MA orders are chosen, typically via brute force search. We attempted to minimize the Akaike information criterion [70] in our search (there are other potential objectives). Next, the time series is fit to the SARIMA model to obtain coefficients for the *AR*, *I*, and *MA* terms.

We performed the first step for several different load profiles to identify reasonable values for the *AR*, *I*, and *MA* model orders, and determined that a SARIMA (1,1,0)$\times$(1,0,0,24) model is appropriate. The non-seasonal component of this model has an auto-regressive order of 1 and an integration order of 1. The seasonal component has an auto-regressive order of 1, and a seasonal period of 24. We fitted all load profiles to this model. The resulting set of coefficients, along with Akaike information criterion values, is visualized in Figure 4.19. Each subplot illustrates the pairwise relationship between two SARIMA

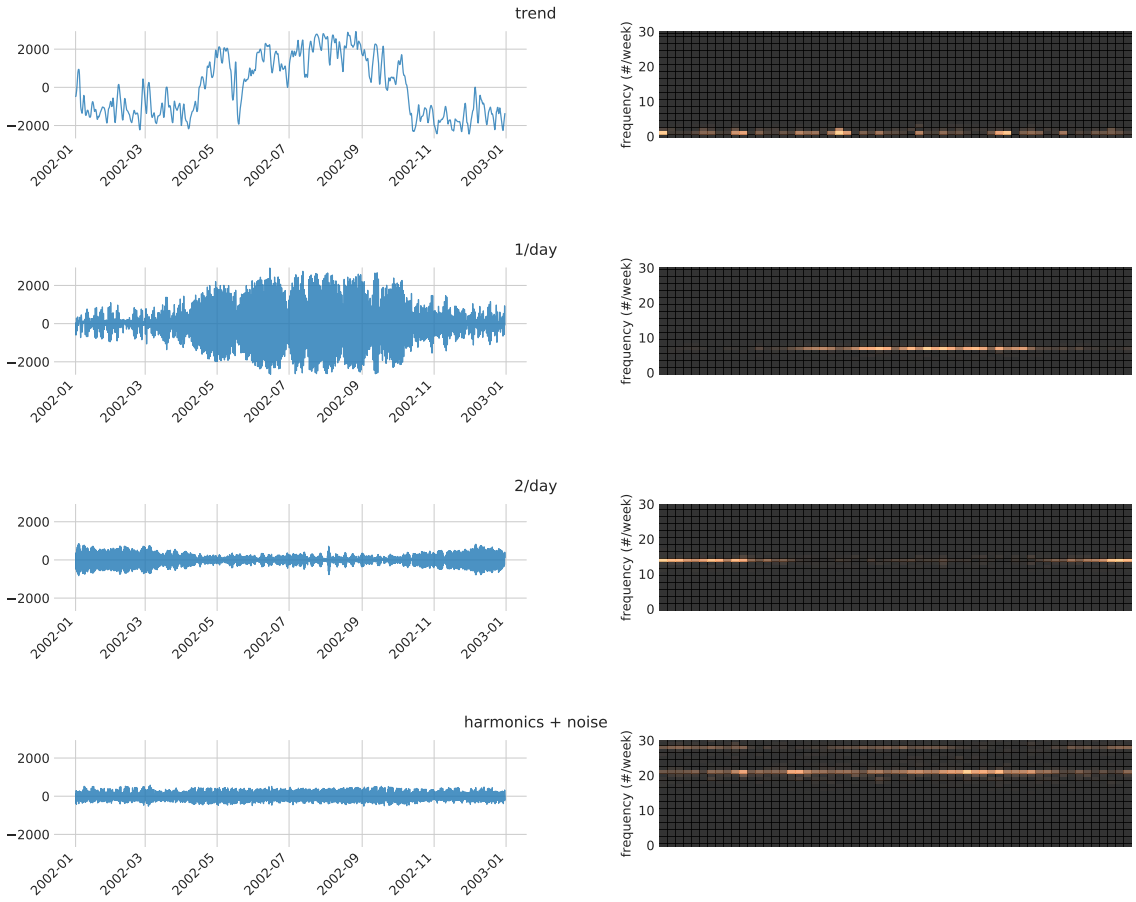


Figure 4.18: SSA components (left) and corresponding spectrograms (right) for one year-long ERCOT load profile. The embedding dimension was chosen to be 24.

model coefficients and the Akaike information criterion. Aside from some clustering (especially for the artificial RTS 96 dataset), there are no obvious patterns that might help to develop a metric. It is also more difficult to explain the physical significance of SARIMA coefficients than, for example, SSA time series components.

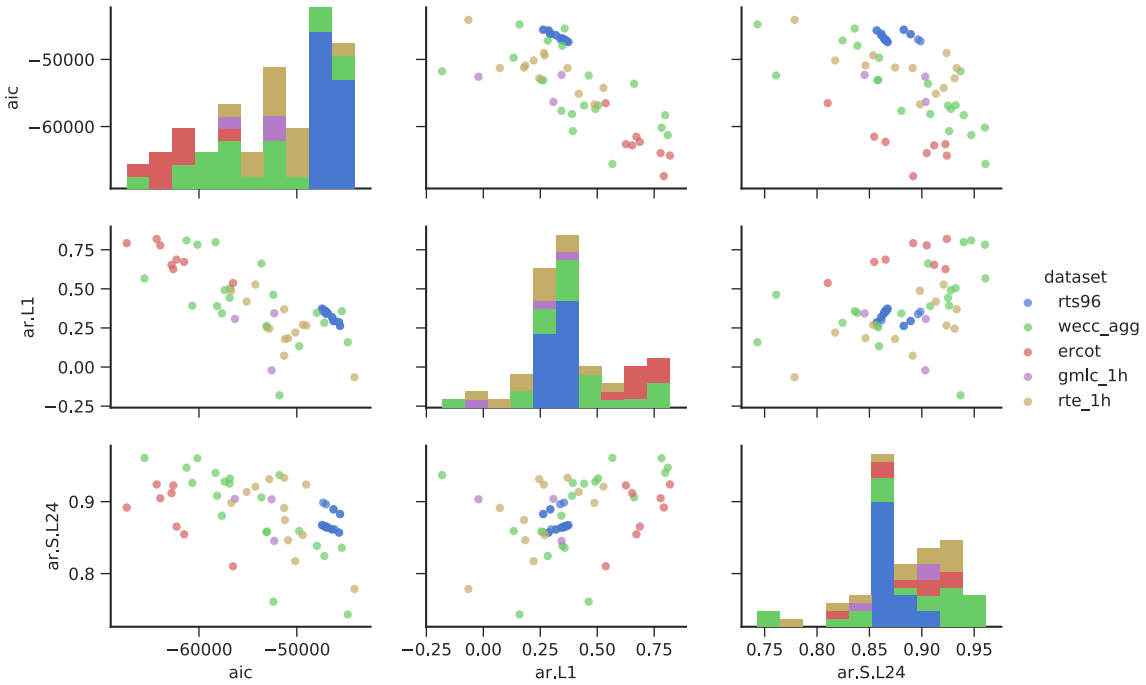


Figure 4.19: Pairplot of coefficients for SARIMA (1,1,0)× (1,0,0,24) model. "aic" is the Akaike information criterion, "ar.L1" is the auto-regressive coefficient, and "ar.S.L24" is the seasonal auto-regressive coefficient.

4.6 Conclusions

This chapter considered six load profile time series datasets. Each was checked for patterns using cross-correlation analysis and singular value decomposition. These methods detect whether a load profile is artificial, and whether disaggregation was performed. We proposed two dataset-wide metrics: one based on the fraction of correlation coefficient values below a threshold close to one, and the other based on data matrix rank. (The latter metric produced unexpected results which will be explained in Chapter 5.) We then applied time series analysis methods to individual time series. Fourier analysis provides an intuitive perspective, but fails to capture the time-varying behavior that makes load profiles interesting. Spectrograms are more appropriate for visualizing electric load profile, though it is unclear

how one might condense spectrogram data into straightforward time series metrics. Singular spectrum analysis can isolate trends, oscillations, and noise with any desired level of granularity. We proposed an embedding dimension and eigen-grouping for quickly decomposing a load profile into four meaningful components. Finally, we investigated SARIMA modeling of load profiles. Our results did not suggest a SARIMA-based metric.

We recommend the following approach for quickly identifying common features present in typical load profile datasets. First, apply the consistency metrics proposed in Section 4.4 to detect re-use of time series across the dataset, and to help determine whether data were generated algorithmically. Next, generate a spectrogram as in Figure 4.15 to identify prominent frequency components and observe their seasonal variation. To investigate further, apply the SSA technique to generate a figure like Figure 4.18. By applying these metrics and analysis methods, one should be able to quickly determine the realism of any load profile dataset.

4.7 Appendix: Importing and Standardizing Datasets

The time series data were stored in various formats, even differing in the character encoding used. We employed Python and the pandas package to import each dataset into a single DataFrame. Three of the datasets –WECC 240, RTS 96, and GMLC– were already available as single files, which made importing straightforward. The ERCOT dataset involves daylight saving time, but was also straightforward to work with. The RTE dataset initially consisted of numerous files spread across multiple directories, and was more difficult to parse into a single DataFrame.

RTE provided four years of 30-minute data for all twelve of France's administrative areas. Each year had a corresponding subdirectory with twelve area-specific data files. These files had a .xls extension, but were in fact TSV (tab-separated variable) files rather than Microsoft Excel workbooks.[4] After programmatically verifying data format consistency across all files, we took advantage of several options in the Python pandas package to correctly parse individual RTE data files. The following notes may be used to reproduce our final tidy RTE dataset.

- Because the original files were saved on an older Windows machine, cp1252 encoding had to be used. (The default encoding for pandas is UTF-8.)
- Turn off the index column and generate it manually later. These files seem to have

[4]This may be verified by opening one of the files in a pager like `less` or a lightweight text editor (each file is approximately 3MB).

delimiters (tabs) at the end of each line, which makes the pandas default index behavior (which designates the first column as the index) undesirable. One must set `index_col=False` to correct for this.

- Recognize 'ND' as a missing value. RTE uses the string 'ND' to represent missing data, so one must set `na_values='ND'` to have pandas recognize this convention.
- Skip the last two rows. Each file in the 2012, 2013, 2014, and 2015 subdirectories ends with two non-data lines. The second-to-last line contains a disclaimer message, and the last line is just a newline. Setting `skipfooter=2` skips the last two rows.
- Merge date and time columns to obtain a timestamp index. The third and fourth columns of each file contain date and time values, so we set `parse_dates=[[2, 3]]` to have pandas combine these columns into a single column of DateTime values. We then use `df.set_index('Date_Heures', inplace=True)` to have pandas use this new DateTime column as the index for the DataFrame.
- Downsample to 30-minute interval. RTE data files have 15-minute resolution to accommodate two columns, but the columns we will analyze contain 30-minute data. Using `df.resample('30T').asfreq()`, we removed the empty rows between data rows.

Combining the aforementioned settings, we obtained a parser for individual data files. We combined the 48 files (12 areas spanning four years) using the following strategy (the code itself is provided in an appendix):

1. For a geographic area (Normandie, for example), find the corresponding file in each year's subdirectory. This produces a list of four file paths.[5]
2. Use the file parser to import the four files into DataFrames.
3. Concatenate these four DataFrames together to obtain a single DataFrame for the area, with an index spanning 2013-2017.
4. Repeat for all other areas.
5. Make a new DataFrame with a hierarchical column index (MultiIndex). The top hierarchical level represents areas, while the second level contains column labels (load, solar generation, etc.).

[5]Unfortunately, some file names (the two corresponding to the Auvergne-Rhône-Alpes and Bourgogne-Franche-Comté areas in each year's subdirectory) were mangled due to the presence of accented characters, which were replaced by URL codes. We changed the filenames to fix this and make it easier to stitch together data from all years for a single area via string matching.

6. Fill this new DataFrame with the twelve individual area DataFrames.

7. Use `DataFrame.swaplevel(axis=1)` to switch the order of the hierarchical levels so the top level represents column labels.

The resulting DataFrame may be indexed using `df['Consommation']` to return all electric load data from 2013-2017 for all 12 areas. The resulting Python pickle occupies 81 MB on disk.

In the end, we obtained a single pickle file for each electric load dataset. These files may be quickly loaded into DataFrames, and all follow one simple format. Indices represent time, while each column contains time series data for one bus or area. Column labels are unchanged from original data sources.

CHAPTER 5

Quantization noise

5.1 Introduction

Electric load time series data are valuable for power systems analysis, particularly in the realms of simulation and planning. System operators and utilities frequently store and share such data with researchers and the public. Load profile datasets typically include several time series (with load aggregated by area), span one or more years, and have time resolution on the order of one hour. This chapter is concerned with two common limitations of such datasets: numerical rounding, also known as *magnitude quantization* and typically performed for storage reasons; and *aggregation*, which represents many small loads using a single time series.[1] Quantization is nonlinear and irreversible, and its effects –referred to as *quantization noise*– are only negligible under certain conditions. When load profile time series are rounded to integer values (an unfortunately common practice), quantization noise is significant enough to interfere with analysis, especially in conjunction with common disaggregation techniques.

Most readily available load profile datasets consist of a handful of aggregated load time series. Aggregation requires fewer measurements, reduces storage requirements, and poses less security risk. Unfortunately, it also limits the data's usefulness to researchers, who often need each load bus in a network to have its own time series. Disaggregation techniques are often used to derive a dataset with one time series per load from a source dataset consisting of aggregated time series. The most common technique is to associate an aggregated load profile with each bus (either at random or based on location), then scale the aggregated time series down according to the portion of load served by that bus. Datasets generated in this manner have the same number of unique load profile shapes as the aggregated load datasets from which they are derived. In other words, if an aggregated

[1] "Quantization" may refer to magnitude quantization (rounding) or time quantization (sampling). Unless otherwise stated, "quantization" refers to magnitude quantization in this chapter.

load dataset and a derived disaggregated dataset are each represented as a data matrix (with each time series as a column), then the two matrices will have the same rank. Unfortunately, the rounding that often follows disaggregation and precedes data storage can obscure this important fact. Quantization noise increases the apparent rank of a disaggregated load profile dataset. The resulting disagreement between reality and numerical appearance can be substantial and confusing.

Using quantization theory, we describe quantization noise in the electric load setting, especially as it influences disaggregated datasets. Relationships between data features, quantization step size, and quantization noise are described. Results are shown for several datasets, illustrating the need to exercise caution when quantizing load profile data. Section 5.2 introduces quantization theory, with standard terminology and relevant results from the literature. This section also introduces time series datasets. Section 5.3 describes the modeling of quantization noise in electric load time series datasets. Section 5.4 presents a numerical encompassing several load profile datasets. Results and conclusions are summarized in Section 5.5.

5.2 Background

Quantization theory dates back to the late 1940s. Bennett's 1948 paper described the relation between distortion and step size for two forms of quantization: rounding (magnitude quantization) and sampling (time quantization) [71]. One key early development was Bennett's assumption (justified under four conditions) that quantization noise may reasonably be modeled as uniformly-distributed, uncorrelated, additive noise. This assumption was popularized by Widrow [72, 73, 74], whose approach substituted distributions of signal values for the signals themselves. Gray summarized the conditions for validity of the uncorrelated additive white noise model as follows: 1) the quantizer does not overload, 2) the quantizer has a large number of levels, 3) the bin width or distance between levels is small, and 4) the probability distributions of pairs of input samples is given by a smooth probability density function [75]. We revisit these conditions in the context of load profile quantization noise later on. Another early and fundamental result is the average distortion for a uniform quantizer [71, 76], restated as (5.5).

Though it is not relevant to the current analysis, another body of quantization literature is worth mentioning. This work traces back to [77], and is concerned with exact descriptions of noise for quantizers driven by structured inputs (a sine wave, for example). While electric load signals contain strong once-per-day and seasonal frequency components, their inherent randomness precludes such exact analysis. Finally, it is appropriate to mention the

random matrix theory that inspires the data matrix analysis in Section 5.3. The thorough and numerically-oriented work [78] is invaluable for those interested in random matrices and their spectra. A quantized disaggregated load profile dataset may also be viewed as a perturbed low-rank matrix, rendering work in this area relevant as well [79, 80].

A quantizer consists of a set of intervals or *cells* $\mathcal{S} = \{\mathcal{S}_i; i \in \mathcal{I}\}$ (where $\mathcal{I}$ is an index set), and a set of reproduction values or *levels* $\mathcal{C} = \{y_i; i \in \mathcal{I}\}$ [81]. Given a real number x, the quantizer output is expressed as $q(x) = y_i$ for $x \in \mathcal{S}_i$, or equivalently

$$q(x) = \sum_i y_i 1_{\mathcal{S}_i}(x) \ , \tag{5.1}$$

where $1_{\mathcal{S}}(x) = 1$ if $x \in \mathcal{S}$ and 0 otherwise. This chapter is concerned with the most common type of quantizer, the *staircase quantizer* associated with rounding. This is a uniform quantizer, with all cells having the same width Δ. If $\Delta = 1$ (integer rounding), we can express q with $\mathcal{S}_i = (i - 1/2, i + 1/2]$, $y_i = i$, and $\mathcal{I} = \mathbb{Z}$. Provided $\mathcal{I} = \mathbb{Z}$ holds, this quantizer will not saturate (or overload), because $\mathcal{S}$ uniformly covers the entire number line. Quantization error, defined as

$$e = q(x) - x \ , \tag{5.2}$$

will therefore lie within $(-1/2, 1/2]$. More generally, a uniform staircase quantizer with cell length Δ has quantization error bounded by $(-\Delta/2,\ \Delta/2]$.

Quantizers are further characterized by rate, distortion, and signal-to-noise ratio (SNR). A quantizer's *rate* captures its storage cost, and is defined as

$$R(q) = \log_2 N \ , \tag{5.3}$$

where N is the number of levels. Quantizer quality is typically assessed via *distortion*. We will use squared error, $d(x,\ \hat{x}) = |x - \hat{x}|^2$, as a distortion measure. With this choice, average distortion $D(q)$ becomes mean squared error (MSE):

$$\begin{aligned} D(q) &= \frac{1}{n} \sum_{i=1}^{n} |q(x_i) - x_i|^2 \\ &= \|e\|^2/n \ , \end{aligned} \tag{5.4}$$

where e is understood to be a vector. One of the earliest and most useful results from quantization theory is the average distortion for a uniform quantizer with high resolution,

due to [71, 76]:

$$D(q) = \frac{\Delta^2}{12} \tag{5.5}$$

Another widely-used result may be derived from (5.5): for a uniform quantizer with high rate, the signal-to-noise ratio (SNR) increases by 6 dB when a bit is added to the rate:

$$\begin{aligned} \text{SNR} &= c + 20R \log_{10} 2 \\ &\approxeq c + 6R \text{ dB} \end{aligned} \tag{5.6}$$

Electric load data are typically made available as collections of aggregated load profiles with a shared time resolution, length, and underlying network. Publicly available load data typically have a time resolution of one hour, and are useful primarily for quasi-static analysis. Let $A \in \mathbb{R}^{n_t \times n_c}$ represent a load profile dataset, where n_c is the number of load profiles, and n_t is the number of samples in each. As an example, the WECC load profile dataset from [82] consists of 21 load profiles spanning one leap year at a one hour resolution; its corresponding data matrix has dimensions 8784×21.

Table 5.1 lists the load profile datasets that will be discussed in Section 5.4. (These are the same datasets that were introduced in Chapter 4.) 'Cols' is the number of time series in the dataset, 'Res.' refers to the interval between timestamps, 'Span' is the time span covered by the dataset, and 'Samples' is the total number of samples in each time series. 'Prec.' is the number of decimal places individual data elements are stored with. There are two variants of the WECC dataset: one with 21 aggregated load profiles from CAISO, and the other derived for use with the WECC 240 network [82] via disaggregation. The disaggregated dataset was generated by assigning each bus in the WECC 240 network a scaled copy of the load profile for the area to which it belongs. Scaling was done according to load participation factors, and the disaggregated dataset was rounded to integer values before being stored. This rounding produced substantial quantization noise, as will be analyzed shortly. The RTS 96 load profile dataset is generated from a set of tables. A scaling factor is associated with each day of the year, hour of the day, and bus in the network. These factors are multiplied to obtain each sample for each bus. As with the disaggregated WECC dataset, RTS 96 load profiles were rounded to integer values before being stored. The resulting quantization noise obscures the low-rank nature of this dataset. The RTE dataset consists of twelve load profiles from RTE in France, each corresponding to one of the country's administrative divisions. The Energy Reliability Council Of Texas (ERCOT) dataset was obtained from the system operator's public website. Each of the eight

Table 5.1: Metadata for load profile datasets.

Dataset	Source	Cols	Res.	Span	Samples	Prec.
Aggregated WECC	[82]	21	1 hr	2004-2005	8,784	1
WECC 240	Disagg. of above	132	1 hr	2004-2005	8,784	0
RTS 96	[83]	51	1 hr	1996	8,736	0
RTE	GRID DATA	12	30 min	2013-2017	80,304	0
ERCOT	[84]	8	1 hr	2002-2018	143,135	13
GMLC	[85]	3	5 min	2020	105,408	7

load profiles corresponds to one of ERCOT's weather zones, which together span most of Texas. The GMLC dataset was synthesized for the grid modernization lab consortium. These three load profiles are associated with a modified version of the RTS 96 network. There is a high degree of aggregation, as each time series corresponds to one area of the RTS 96 network.

5.3 Error modeling and signal recovery

The additive white noise model is reasonable for quantized load profile datasets, as the four assumptions involved are typically valid. First, the rounding quantizer does not overload (there is no cell with width greater than Δ). Second, the quantizer has a large number of levels, since electric load is typically reported in MW with values on the order of 10^3 (though we will consider scenarios where this is not the case). Third, the cell width Δ is small relative to the size of the signal, as it typically lies in $[10^{-6},\ 1]$. The fourth assumption, pertaining to smooth distributions of pairs of signal values, may be justified empirically by considering a load profile's distribution of values. For a time series with values on the order of 10^3 and integer precision or greater, probability densities of elements are empirically smooth. The additive white noise model may therefore be used to estimate or bound the contribution of quantization noise to a quantized time series. The vector case (one load profile) is straightforward, and leads naturally to the matrix case (multiple load profiles).

Consider first the case where a single load profile is represented as a vector of length n_t. Let $q(\cdot)$ be the rounding quantizer with step size Δ, and $q.(\cdot)$ denote element-wise application of q to an array argument. Now let $z_1 \in \mathbb{Z}^{n_t}$ be an integer-valued vector, and $\alpha \in \mathbb{R}$ a scalar. Then $z_2 = q.(\alpha z_1)$ is a scaled and quantized version of z_1, and the quantization error vector is

$$e = z_2 - \alpha z_1.$$

Now assume Bennett's model of uniformly distributed and uncorrelated quantization noise

is valid, and Δ is "small enough" relative to αz_1. Distortion is estimated using (5.4) and (5.5) as

$$D(q) = \frac{\|e\|^2}{n_t} = \frac{1}{n_t}\sum_{i=1}^{n_t}|e_i|^2 \tag{5.7}$$

$$\approx \frac{1}{n_t}\sum_{i=1}^{n_t}\frac{\Delta^2}{12} \tag{5.8}$$

$$\implies \|e\|^2 \approx n_t \cdot \frac{\Delta^2}{12} \tag{5.9}$$

$$\|e\| \approx \Delta\sqrt{\frac{n_t}{12}}\,, \tag{5.10}$$

where in (5.8) we used (5.5) (and the assumption that Δ is small) to replace $|e_i|$ with its average value. To add numerical context: a quantized load profile spanning one year at a one-hour resolution ($n_t = 8760$) has $\|e\| \approx 27\sqrt{\Delta}$ according to (5.10). In other words, quantization error effectively adds a noise vector with uniformly random values, and a norm of roughly $27\sqrt{\Delta}$.

We next consider multiple time series together. A load profile dataset consists of multiple time series, and may be represented as a matrix $A \in \mathbb{R}^{n_t \times n_c}$. If the rounding quantizer q is applied to A, and the additive white noise model is valid, then the error matrix may be expressed as

$$E = q.(A) - A\,. \tag{5.11}$$

To clarify subsequent notation, let $A_q = q.(A)$. While it is not possible to recover A from A_q in general, there is a common setting where the low-rank nature of A permits signal recovery from A_q. Consider an aggregated load profile dataset $A \in \mathbb{R}^{n_t \times n_a}$ consisting of n_a linearly independent time series. Now suppose a disaggregated dataset $D \in \mathbb{R}^{n_t \times n_d}$, $n_d > n_a$, is derived from A, such that each column of D is a linear combination of columns of A. Without loss of generality, assume that $\text{rank}(D) = n_a$ (implying that all columns of A are used in the construction of D). Then when D is quantized, the singular value

decomposition may be used to isolate quantization noise as follows:

$$\begin{aligned} q.(D) = D_q &= D + E = USV^\top \\ &= \begin{bmatrix} U_D & U_E & U_0 \end{bmatrix} \begin{bmatrix} S_D & 0 & 0 \\ 0 & S_E & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_D \\ V_E \\ V_0 \end{bmatrix} \\ D_q &= U_D S_D V_D^\top + U_E S_E V_E^\top \ . \end{aligned} \tag{5.12}$$

This step requires singular values associated with signal to be greater than those due to noise, but it is straightforward to show that this requirement is satisfied in every case where the additive noise model is valid.[2] According to (5.12), D may be recovered approximately from D_q by summing the first $\text{rank}(D) = n_a$ components of the singular value decomposition of D_q. To summarize: when a disaggregated load profile dataset D is derived from an aggregated source dataset A having known rank n_a, and D is then quantized, one may approximately recover D via the singular value decomposition of $q.(D)$.

Suppose now that the aggregated source dataset is unavailable, and its rank n_a unknown. It is possible to determine n_a from D_q alone by estimating a singular value threshold to separate signal and noise components (provided, again, that quantization noise is not large enough to overwhelm the signal). Then n_a is simply the number of singular values above the threshold, with the remaining components comprising quantization noise. The remainder of this section derives an appropriate singular value threshold, and the idea is explored numerically in Section 5.4.

The problem of determining n_a given D_q is equivalent to the estimation of $\max(S_E) = \|E\|_2$, the largest singular value (or spectral norm) of the error matrix E. This value is bounded trivially as follows:

$$\|e_j\|_2 \leq \|E\|_2 \leq \|E\|_F \quad \forall\, j, \tag{5.13}$$

where e_j is the j-th column of E, and $\|\cdot\|_F$ is the Frobenius norm. As with the single load profile case, assume each element of E is sampled from a uniform distribution over the range $(-\Delta/2,\ \Delta/2]$. The bounds may be re-stated using the sample standard deviation σ_e [86]:

$$\sqrt{n_t}\sigma_e \leq \|E\|_2 \leq \sqrt{n_t n_d}\sigma_e. \tag{5.14}$$

[2] It is worth pointing out that the additive noise model is never exact. Quantization noise is deterministic, and is therefore correlated with the signal, even if only slightly.

Provided Δ is small relative to the signal, the variance is given by (5.5), and therefore[3]

$$\sigma_e = \frac{\Delta}{\sqrt{12}}.$$

Substituting into (5.14), we obtain readily computable bounds:

$$\sqrt{\frac{n_d \cdot \Delta}{12}} \leq \|E\|_2 \leq \sqrt{\frac{n_t n_d \cdot \Delta}{12}}\ . \tag{5.15}$$

Provided the underlying signal is not overwhelmed, SVD components with singular values between these bounds are due to quantization noise. This suggests a straightforward means of isolating and subtracting quantization noise in load profile datasets generated via disaggregation. We turn next to a numerical treatment of several datasets.

5.4 Numerical study

5.4.1 Singular value gaps for WECC and RTS-96

Two of the datasets in Table 5.1 are (mathematically, at least) low-rank, and serve as excellent examples of quantization noise effects. The WECC disaggregated dataset is derived from a set of 21 aggregated load profiles, and the RTS-96 dataset is derived from one function. The datasets should have ranks 21 and 1, respectively, but both were integer-rounded prior to storage; this quantization causes numerical reality to differ from mathematical expectation. Figure 5.1 depicts singular values of the WECC and RTS-96 data matrices. The former has numeric rank 128; the latter 17. Each spectrum has a noticeable gap (after the 20th singular value for WECC, and 1st singular value for RTS-96), along with numerous densely-packed values in a plateau of sorts. This suggests that each data matrix consists of a signal rising above a "noise floor". From the analysis in Section 5.3 and the dashed-line bounds in Figure 5.1, the presence of quantization noise is already apparent, but it is worthwhile to go through the process of identifying quantization noise unaided by bounds.

The left and right singular vectors illustrate how components above the gap differ fundamentally from those below. Consider the fourth left singular vector U_4 of the WECC dataset, shown at the top of Figure 5.2. This signal has a periodic makeup, and the cor-

[3]The standard deviation may also be derived from the uniform distribution density:

$$\sigma_e = \frac{1}{\sqrt{12}}(b-a) = \frac{1}{\sqrt{12}}[0.5\Delta - (-0.5\Delta)] = \frac{\Delta}{\sqrt{12}}$$

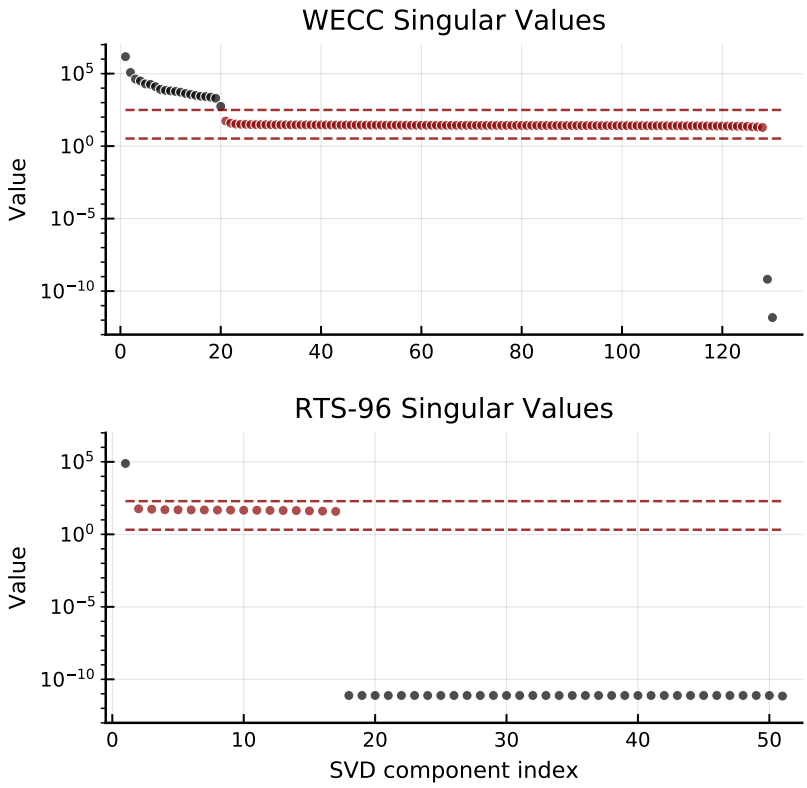


Figure 5.1: Singular values for WECC (top) and RTS-96 (bottom) time series data matrices. Error bounds are shown with dashed red lines, and singular values between these bounds are colored to match.

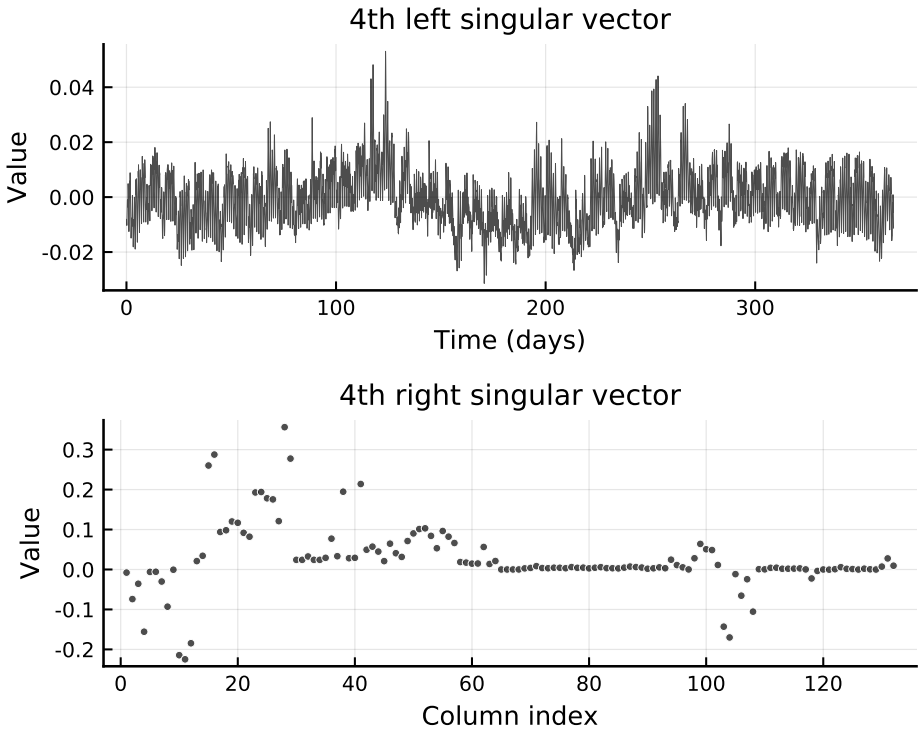


Figure 5.2: 4th singular value component for WECC dataset. Left singular vector on top; right singular vector on bottom. The corresponding singular value, σ_4, is 31,079.

responding right singular vector V_4 (bottom of Figure 5.2) shows that U_4 is an important ingredient for many of the 132 columns.[4] This is representative of components 1-20. By contrast, the 21st left singular vector (top of Figure 5.3) appears to be noise[5], and the corresponding right singular vector (bottom of Figure 5.3) shows that U_{21} only plays a significant role in a few of the disaggregated WECC time series. We conclude that U_{21} exists primarily to represent quantization noise for a few columns of the data matrix.

The bounds derived in Section 5.3 prove useful for the disaggregated WECC and RTS-96 datasets. For WECC, $n_t = 8784$, $n_d = 132$, and $\Delta = 1$. Substituting into (5.15) yields

$$3.33 \leq \|E\| \leq 312.02 \, .$$

These bounds are illustrated with dashed red lines in Figure 5.1, and suggest that there are 20 signal components: the 20th singular value ($\sigma_{20} \approx 554$) is above the upper bound, while the 21st ($\sigma_{21} \approx 54$) is below it.[6] For the RTS-96 dataset, $n_t = 8736$, $n_c = 51$, and the

[4]Each element $V_4[i]$ captures the relative importance of U_4 in representing column i.

[5]This is readily confirmed via FFT analysis.

[6]Given that the disaggregated WECC dataset was derived from 21 load profiles (not 20), this result is somewhat surprising. The discrepancy is explained by excessive signal scaling later in this section.

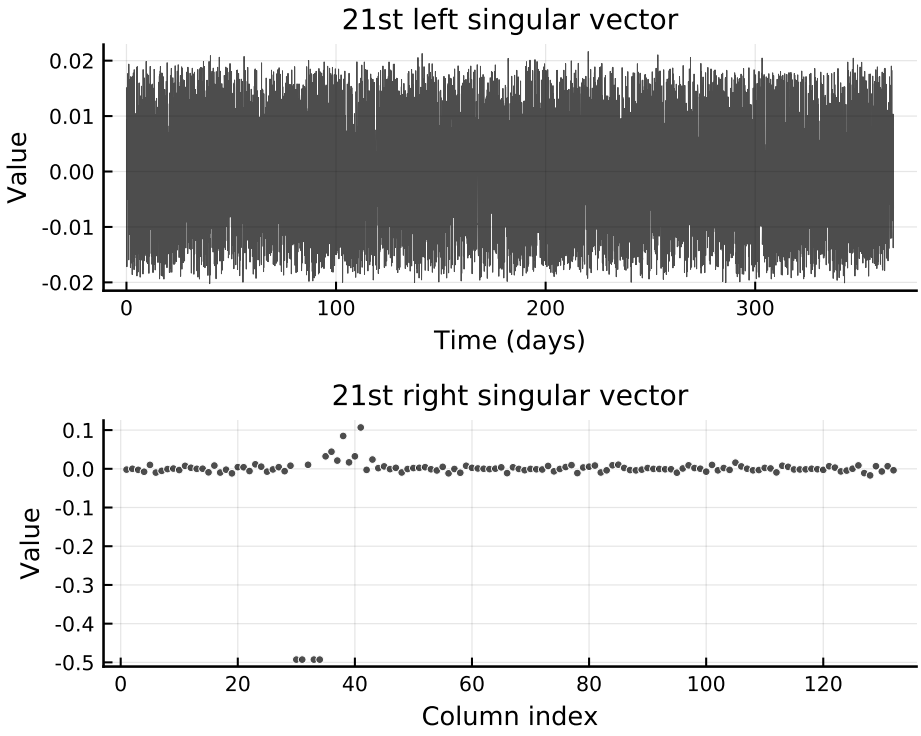


Figure 5.3: 21st singular value component for WECC dataset. Left singular vector on top; right singular vector on bottom. The corresponding singular value, σ_{21}, is 54 (three orders of magnitude below σ_4).

bounds become

$$2.08 \leq \|E\| \leq 194.57 \, .$$

These bounds are depicted in the bottom subplot of Figure 5.1, and suggest there is one signal component. This matches prior knowledge of the dataset, each of whose time series was derived from the same function. While the error matrix bounds do prove useful for the WECC and RTS-96 datasets, it is worth noting that determination of the true rank of a matrix with additive white noise is difficult in general [86]. Quantization noise is inherently correlated with the signal, even if only slightly. The assumption of many signal levels breaks down for excessively-scaled time series, as will be discussed shortly.

5.4.2 Ranks of WECC and RTS-96 data matrices

If each column of these matrices has a unique quantization noise pattern, one would expect full rank. It is noteworthy, then, that neither the WECC data matrix nor the RTS-96 data matrix has full numerical rank. A closer look at the WECC and RTS-96 datasets reveals that some columns are indeed scaled similarly enough to share a quantization noise pattern, resulting in a lower-rank E matrix. This leads to the following observation: for two quantized columns to be linearly dependent, they must 1) be derived from the same non-quantized time series (or same linear combination of non-quantized time series); and 2) must be scaled such that their quantization noise patterns align. If two quantized time series have the same pre-quantization shape but are scaled differently, the difference in quantization error will require an additional linearly independent vector to capture. The rank of a data matrix is a function both of unique shapes and various quantizations thereof. For each unique shape represented, there is one singular value above the noise threshold. Differently-scaled versions of these time series have various quantization noise patterns (which may align); for each of these patterns, there is a singular value below the noise threshold whose corresponding component helps to offset it.[7]

The WECC data matrix contains just five numerically identical columns, resulting in a 4-dimensional nullspace. For the RTS-96, there are 17 differently-scaled and quantized versions of one load profile, corresponding to the 17 buses in each of the network's three identical areas. Each of these columns has its own scaling factor [83] and quantization noise pattern. This explains the large gap between σ_1 and σ_2, as well as the 16 singular values within error bounds.

[7] There is not, of course, a one-to-one correspondence between each quantization noise pattern and one component of E; the components of E work together to offset the full effects of quantization for all columns.

5.4.3 Scaling and signal degradation

Quantization noise becomes more significant as the input signal is scaled down. To render quantization error negligible, a signal must be significantly larger than cell width Δ, and capable of taking many unique values after quantization. When a time series is scaled down excessively before quantization, the resulting signal may be dominated by noise, resulting in choppy and unrealistic load profiles.

The deleterious effects of scaling and quantization are best expressed with an example. Let x be one week of load data (at a one-hour resolution) from a WECC time series, and $q(\cdot)$ be the integer-rounding quantizer. Now let

$$y_1 = q(x)$$
$$y_2 = q(x/50)$$

be two differently-scaled and quantized versions of x. Figure 5.4 illustrates y_1 (top) and y_2 (bottom). The original signal x takes 147 unique values. Despite quantization, y_1 is still large enough to take 61 unique values, and quantization error (shown in gray) is relatively small. By contrast, y_2 was squeezed so much before quantization that it can only take three unique values; quantization error is on the same order of magnitude as y_2 itself. To generalize slightly, let x be a load profile, and α a scaling factor. Define $u(x)$ to be the number of unique values in x, i.e.

$$u(x) = |\text{unique}(x)|, \tag{5.16}$$

where unique$(\cdot)$ returns the set of unique values of its argument. It is interesting to consider the relationship between α and $u(q(\alpha x))$ (the number of unique values αx takes after quantization).

Figure 5.5 was obtained by plotting $u(q(\alpha x))$ against α for the 21 load profiles in the aggregated WECC dataset. Note that each trace becomes flat past a certain value of α (either $\alpha = 1$ or $\alpha = 10$ for this dataset). This flattening is a function of the number of unique values in the non-quantized time series; a signal with k unique values cannot have more than k values when it is scaled up, but it will have fewer than k unique values when scaled down. Thus, this flattening behavior reveals the true precision of a time series. Figure 5.5 also provides context for Figure 5.4, whose blue and red traces correspond to the blue and red circles. The time series used to generate Figure 5.4 is particularly susceptible to down-scaling degradation due to its low peak value. The 132 green points in Figure 5.5 illustrate how the 132 columns of the disaggregated WECC dataset were derived from

the 21 aggregated time series. These points were obtained from the process described in Appendix I. While most of the points correspond to reasonably large numbers of unique values, a handful lie in the 3-10 range. Quantization error for these signals will be relatively large, and they will appear unrealistically choppy (like the bottom plot in Figure 5.4).

Figure 5.6 is generated in the same manner, but includes one time series from each dataset listed in Table 5.1. It is immediately obvious that the ERCOT time series (trace A) is the only one that was stored at floating-point accuracy; its trace is the only one that does not flatten off. The aggregated WECC time series (trace D) flattens off at $\alpha = 10$ (implying single decimal place precision), while the RTE and RTS 96 time series (traces C and E) have integer precision. Based on its seven decimal places, one would expect the GMLC time series (trace B) to taper off gradually like the ERCOT trace; instead, it flattens at $\alpha = 3.08$. We conclude that the GMLC time series is not free to take all seven-digit-precision values between its extrema. Further inspection reveals it is in fact limited to 5365 unique values. Sorting these values and taking the first difference shows that almost all are spaced by 0.3243 MW. As expected from Figure 5.6, $1/0.3243 = 3.08$, which matches the value of α at which trace B flattens.

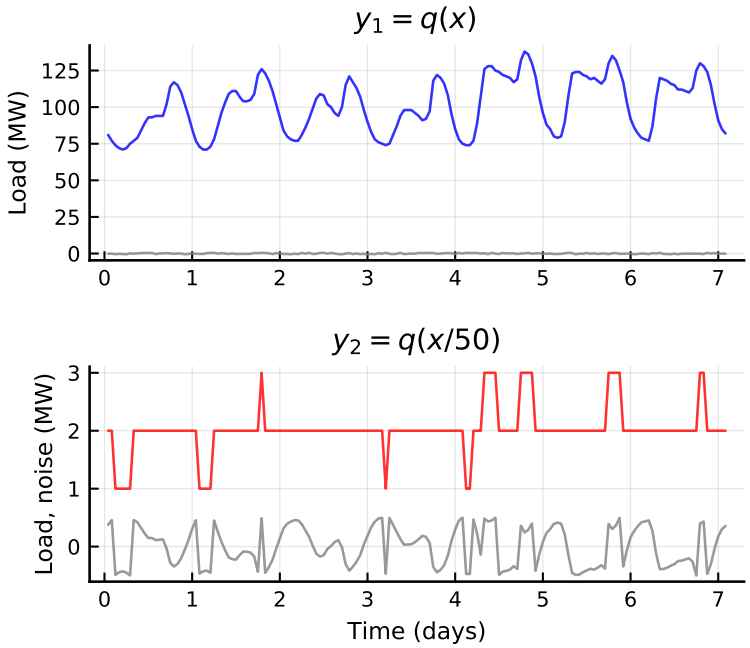


Figure 5.4: Quantization noise dominates signals that are scaled down too much. These two differently-scaled versions of the same WECC time series, for example, look quite different after quantization. Quantization error, shown in gray, is much more significant in the bottom subplot.

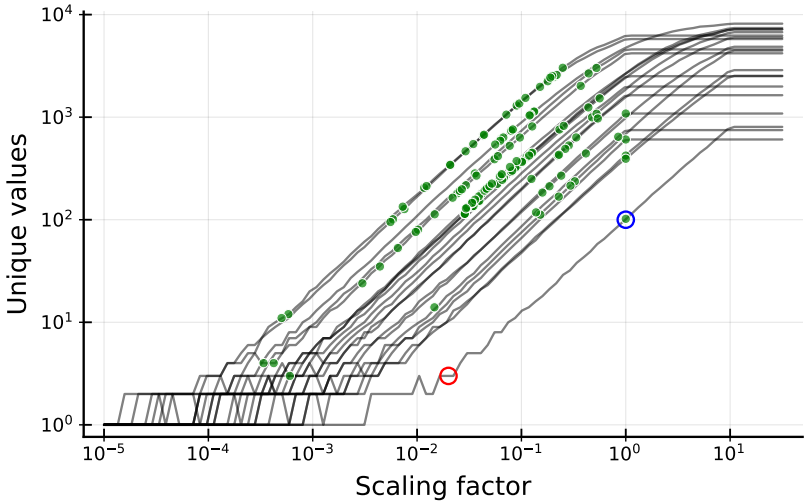


Figure 5.5: Number of unique elements in quantized signal versus scaling factor for all 21 load profile time series from the aggregated WECC dataset. Note the blue and red circles, which correspond to the blue and red (top and bottom) plots in Figure 5.4. The green circles illustrate the derivation of the disaggregated WECC dataset.

5.5 Conclusions

Quantization noise is nonlinear and irreversible, yet detectable and predictable. We have described the validity of the additive white noise model in the load profile setting, and have used this model to derive error estimates and bounds that are both useful and straightforward to compute. Using these bounds, one can readily identify quantization noise, and even subtract it from a load profile data matrix derived via disaggregation from a low-rank aggregated data matrix. Numerical analysis illuminates the difference between signal and quantization noise singular value components. The relationships between scaling, the number of unique elements a signal can take, and quantization noise were also considered. We provide a satisfactory explanation of the "shelf" of singular values between the data components and the noise floor observed in quantized time series datasets. It is our hope that the reader stands better equipped to identify quantization noise in their own time series data, and to avoid introducing quantization noise where possible. Quantization noise may be avoided if time series data are simply stored with additional precision to minimize quantization noise. Each additional bit of precision shifts the noise floor down by 6 dB [71], and modern data storage resources make it inexpensive to store large time spans of hour-resolution data to several decimal places.

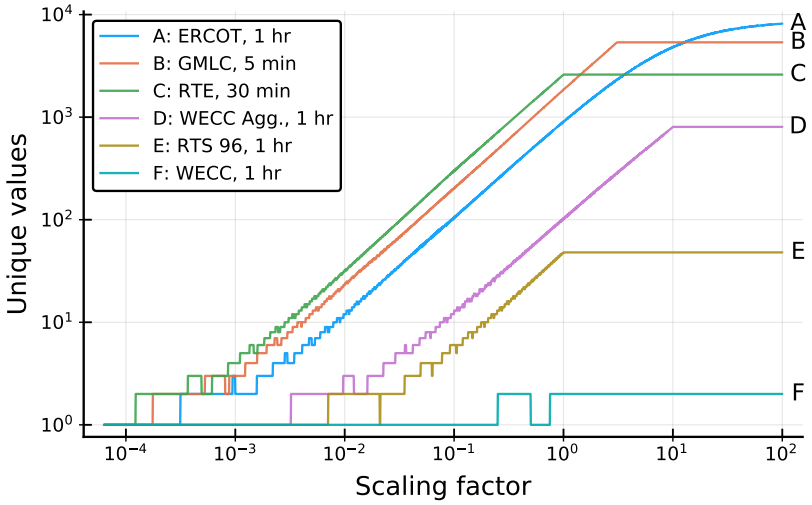


Figure 5.6: Number of unique elements versus scaling factor for one integer-rounded time series from each dataset (selected according to lowest peak value). Each time series was shortened to one year.

Appendix: Deriving a Floating-point Version of WECC Data

Having access to both aggregated and disaggregated WECC datasets, one can derive a floating-point variant of the disaggregated dataset. Let A represent the aggregated WECC dataset (21 integer-valued columns), and D represent the disaggregated WECC dataset (132 columns, significant quantization error). Assuming each column of D is a scaled (and quantized) version of just one column of A (which we know to be the case), we can write

$$d_i = q(\alpha_i a_j),$$

for each column d_i of D, where $q(\cdot)$ is the integer-rounding quantizer. The following procedure may be used to determine the correspondence between column indices i and j, and scaling factors α_i.

1. Normalize all columns of A and D to obtain $\bar{A}$ and $\bar{D}$.
2. For each column $\bar{d}_i$, find the column $\bar{a}_j$ that minimizes $\|\bar{d}_i - \bar{a}_j\|$. This yields the index j of the aggregated load column that was scaled to obtain d_i.
3. Now determine α_i by taking the mean of the element-wise division d_i/a_j.

Figure 5.7 shows the singular values for the floating-point disaggregated WECC dataset derived in this manner. It is now reasonable to claim that the matrix has rank 21, which is equal to the number of aggregated load profiles as mathematically expected. The "noise

floor" due to quantization has dropped to the level of floating-point accuracy. Due to the nature of floating-point arithmetic, the difference ϵ between consecutive values is not constant, and one cannot determine a fixed Δ. One choice of error bounds that encapsulates the noise floor (and is shown in Figure 5.7) corresponds to $\Delta = 10^{-23}$, which is the difference between two floating-point values at 10^{-7}.

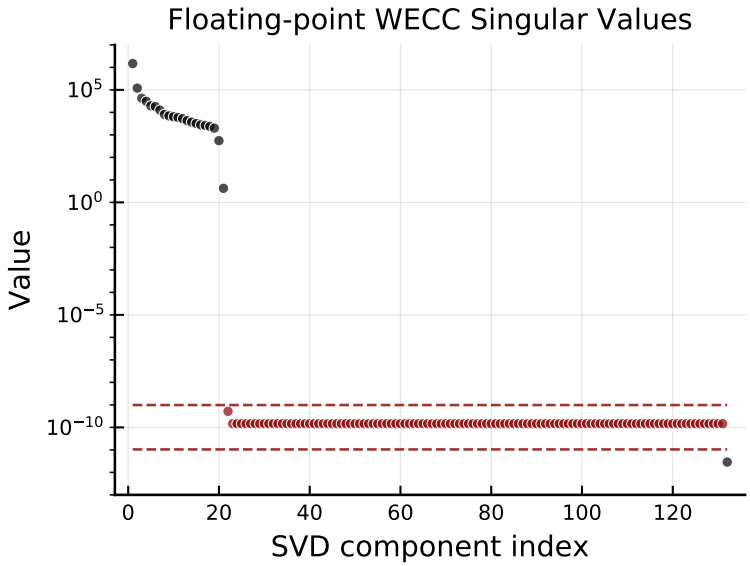


Figure 5.7: Singular values for floating-point WECC. Error bounds (dashed red lines) are derived assuming $\Delta = 10^{-23}$.

CHAPTER 6

System structure and reliability

6.1 Introduction

The buses and transmission lines of a power grid translate naturally to the nodes and edges of a graph. This connection has been recognized for many years, and numerous graph structural properties have been studied with various power systems applications [87, 88]. Graph-theoretical methods have been used to identify system vulnerabilities [8, 9, 10], detect structural anomalies [11], generate and validate synthetic grids [12, 13, 14, 15], create meaningful visualizations [56, 89], and perform partitioning [90, 91]. The graph analysis methods employed may be divided into two categories: weighted, where electrical information is embedded in the graph, and unweighted, where only topology is considered. Though there is a fundamental difference between a power system's topology and its electrical structure [9, 92, 13], unweighted graph analysis is ideal for quickly detecting unusual connectivity patterns. The topological algorithms used in this chapter are computationally inexpensive, and interpretation of numerical results is straightforward. Consequently, it is feasible to scan large power grids to check for structural anomalies.

Efficient detection of structural anomalies calls for two ingredients. The first is a large collection of test networks from which a notion of "normal" topology may be derived. NESTA [28] is well-suited for this purpose. The second ingredient is a combination of straightforward, computationally inexpensive graph metrics capable of highlighting deviations from the norm. One of the main contributions of this chapter is selection of several topological metrics (along with discussion of less useful ones that were also considered), their application to NESTA, and revelation of structural anomalies. Our methods efficiently reveal highly-connected subnetworks in all four PEGASE project [93] topologies, in addition to highlighting the unusual density of the 162-bus IEEE dynamic test case.

The second portion of this chapter expounds on the connection between maximal cliques and a recently-developed semidefinite programming (SDP) optimal power flow (OPF) al-

gorithm [16]. Because SDP solver time increases sharply with semidefinite constraint matrix size, decomposition into a set of constraints with smaller submatrices can significantly improve solver performance [94]. According to a matrix completion theorem, any valid decomposition must consist of combinations of maximal cliques. We study the greedy clique merge algorithm proposed in [16], which merges overlapping cliques in search of a decomposition that minimizes SDP solver time. Results show that solution time is not minimized at the point one might expect, suggesting that topological considerations merit further study in the SDP OPF context.

The chapter proceeds as follows. Section 6.2 introduces graph notation and defines various topological graph metrics. Section 6.3 applies these metrics to NESTA networks, characterizing typical grid topology and highlighting unusual connectivity patterns. Section 6.4 presents conclusions.

6.2 Metric definitions and intuition

A power grid consists of electrical generators and loads joined by power lines and transformers. The structure of a power grid may be represented as a graph, which consists of a set of nodes $\mathbf{N}$ connected by a set of edges $\mathbf{E}$. Connections may be encoded in an adjacency matrix A, where each element A_{ij} is 1 if nodes i and j have an edge between them, and 0 otherwise. The translation from power grid to graph involves a few modeling choices. We consider only undirected, unweighted graphs, where edges have no orientation or other properties. In the test networks we consider, each substation is one node. Though the data has this limitation, our ideas are applicable to high-fidelity models of real grids, which often have multiple buses within a substation [89]. We omit parallel edges between vertices. Finally, because some public test networks use transmission line objects to model transformers (and vice versa in rare cases), we allow both power lines and transformers to be edges in our graph representations.[1] The remainder of this section defines several topological graph metrics we will subsequently apply to numerous test networks.

[1]This can add a substantial number of "leaf nodes," but we repeated all analysis with generator step-up transformers excluded to verify that results are not qualitatively affected.

6.2.1 Degree distribution

In a graph, each node's degree is the number of nodes to which it is connected. Mathematically, the degree of node i in a graph with $|\mathbf{N}|$ nodes is given by

$$k_i = \sum_{j=1}^{|\mathbf{N}|} A_{ij} \ . \tag{6.1}$$

The node degree distribution $\mathrm{P}(k)$ is a meaningful and concise metric based on this property:

$$\mathrm{P}(k) = \frac{|\mathbf{N}_k|}{|\mathbf{N}|} \ , \tag{6.2}$$

where $\mathbf{N}_k$ is the set of nodes with degree k [13]. Degree distribution has system reliability implications [9, 15, 89, 14, 10] and reveals some useful information at a glance. A long tail with high-degree nodes indicates one or more strongly-connected hub nodes (typically at high voltage levels [89]), which make the system more vulnerable to targeted attack [8, 10]. A high relative density of degree-1 and degree-2 nodes suggests a grid with long paths or radial components, and low meshing [10]. We have found that maximum, mean, and median values sufficiently highlight most interesting features and anomalies of NESTA networks. Maximum and mean degree values are defined, respectively, as:

$$k_{\max} = \max k : |\mathbf{N}_k| > 0 \tag{6.3a}$$

$$\bar{k} = \frac{1}{|\mathbf{N}|} \sum_{k=1}^{k_{\max}} k \cdot |\mathbf{N}_k| \tag{6.3b}$$

Median node degree k_{med} is defined in the usual way; it is the middle value of the sorted set of node degrees (or mean of the two middle values if the number of nodes is even).

Considerations of flexibility, reliability, and physical space keep $k_{\max}$ from growing too large in real power grids. In the North American power grid, the probability of a substation having more than x transmission lines decreases exponentially with x [10]. Thus, high $k_{\max}$ hints that network reduction or perhaps even a modeling error has occurred. The mean distribution value $\bar{k}$ is related to how meshed the system is, and lies between 2 and 3 for most grids. Median node degree is difficult to find in the literature, and is less useful for characterizing system structure. Though some small, dense networks have median degree 3, the vast majority have $k_{\text{med}} = 2$.

6.2.2 Degree assortativity

The degree assortativity coefficient r measures the extent to which nodes of like degree connect to each other. It is equivalent to the Pearson correlation coefficient of node degrees [95]. If $\mathbf{K} = \{k : |\mathbf{N}_k| > 0\}$, then r is given by

$$r = \frac{1}{\sigma_k^2}\left[\sum_{x,y\in\mathbf{K}} xy\left(\frac{|\mathbf{E}_{xy}|}{|\mathbf{E}|} - \frac{|\mathbf{E}_x|}{|\mathbf{E}|}\frac{|\mathbf{E}_y|}{|\mathbf{E}|}\right)\right]. \tag{6.4}$$

In (6.4), x and y represent any two node degree values, $|\mathbf{E}_{xy}|/|\mathbf{E}|$ is the fraction of all edges connecting a node with degree x to a node with degree y, $|\mathbf{E}_x|/|\mathbf{E}|$ is the fraction of all edges that start or end at nodes with degree x, and σ_k is the standard deviation of the node degree distribution $\mathrm{P}(k)$. Values of r range from -1 to 1, indicating perfect disassortativity and assortativity respectively [95]. Power system topologies tend to be slightly disassortative, as mentioned by [15, 13] and confirmed in Section 6.3.2.

6.2.3 Rich-club coefficient

High degree assortativity suggests a strongly-interconnected set of high-degree nodes. The rich-club coefficient [96] detects this structure, sometimes referred to as a "hubs of hubs" or "rich club," directly. Let $\mathbf{N}_{>k}$ be the set of nodes with degree greater than k, and $\mathbf{E}_{>k}$ the set of edges between those nodes. Then the rich-club coefficient is computed for each degree k as

$$\phi(k) = \frac{2|\mathbf{E}_{>k}|}{|\mathbf{N}_{>k}|\left(|\mathbf{N}_{>k}| - 1\right)}, \tag{6.5}$$

which is the fraction of all possible edges that exist among nodes with degree greater than k. If there is some degree $x : \phi(k) = 1\ \forall\ k > x$, then ϕ is said to "saturate" [11], and the graph is said to exhibit the "rich-club effect" [96]. Only two nodes and one edge are required to make this happen: if the two highest-degree nodes in a graph each have degree y and are connected, with the next-highest node degree being x, then $\phi(k) = 1\ \forall\ k \in [x, y-1]$. It is therefore important to consider the number of nodes involved in a rich club to ensure one is not simply observing two highly-connected nodes.

6.2.4 Cliques

A fully-connected subgraph is called a clique. In graph notation, a clique $\mathbf{C}$ satisfies

$$\mathbf{C} \subseteq \mathbf{N} : A_{ij} = 1 \; \forall \; i, j \in \mathbf{C} \; . \tag{6.6}$$

Although each subset of a clique is technically also a clique, we restrict our focus to those *maximal* cliques which cannot be expanded by adding other nodes. Determining the maximal cliques of a general undirected graph is an NP-hard problem, but a memory-efficient adaptation of the Bron-Kerbosch algorithm [97] identifies the maximal cliques of our NESTA power grid graphs in a reasonable time.

6.2.5 Chordal graph extensions

A graph is chordal if every cycle of at least four nodes has a chord, which is an edge between nodes that are non-adjacent in the cycle. While power grid graphs are not chordal in general, it is straightforward to obtain a chordal extension of a non-chordal graph via sparsity-preserving Cholesky factorization of the graph adjacency matrix [94]. This yields a chordal graph with a few new edges. While identifying all maximal cliques of a general graph is NP-hard, there is a linear-time algorithm that applies to chordal graphs [98]. Chordal graph extensions and their cliques play a key role in improving SDP OPF performance, as will be shown in Chapter 7.

6.2.6 Adjacency spectral radius

Adjacency spectral radius refers to the largest eigenvalue of the graph adjacency matrix A. The eigenvalue spectrum of A is expensive to compute relative to aforementioned metrics, especially for larger networks. While this metric yields interesting results, computation time makes it less useful for quickly revealing structural anomalies. In Section 6.3.5 we show that a combination of other metrics reveals similar information more quickly.

6.3 Structural anomalies in NESTA networks

Of the thirty power system graph analysis papers reviewed in [88], nearly 85% use one of the IEEE Literature-Based Power Flow Test Cases (e.g. the IEEE 14-Bus System). NESTA [28] includes these networks, RTE and PEGASE systems [99, 93], and many other publicly-available test networks. This makes NESTA an ideal proving ground for graph

metrics. Because the archive includes multiple versions of some topologies (e.g. several Polish grid operating conditions), we focus on the subset of 33 representative NESTA networks shown in Figure 6.1 to ensure no single topology is over-represented. In the remainder of this section, we apply the metrics described in Section 6.2 to this collection of networks. Any graph analysis that was not hand-coded was performed by NetworkX [100].

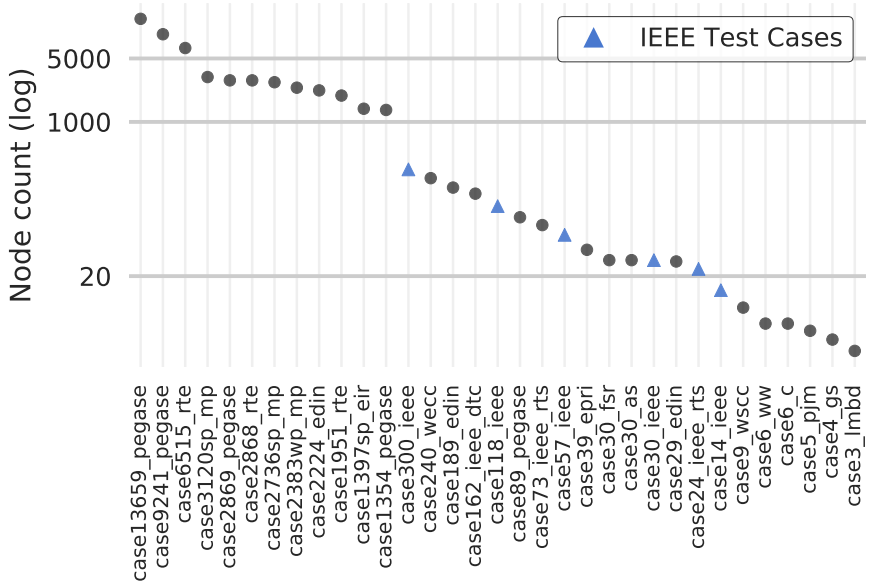


Figure 6.1: Semilog plot of node counts, illustrating size categories.

6.3.1 Degree distribution

A graph's maximum node degree is the highest degree observed among its nodes (6.3a). Figure 6.2 shows that this property scales roughly linearly with the log of network size for NESTA networks. Although the trend makes it appear that maximum node degree can grow arbitrarily large, it is important to note that real substations are limited by the design constraints discussed in Section 6.2.1. The three labeled outliers in Figure 6.2 are all PEGASE networks, and the three points with maximum degree 9 in the 1,000-5,000-node size range are the three Polish grid variants included in our NESTA sample. The unusually high maximum degree of the large PEGASE topology has been pointed out [11], but to our knowledge no prior work has shown how unusual the 89-bus PEGASE system is for its size. This network's highly-connected hub component is apparent from visual inspection of a 2D graph layout, but maximum node degree and other metrics discussed later on can identify this abnormal structure more quickly and efficiently.

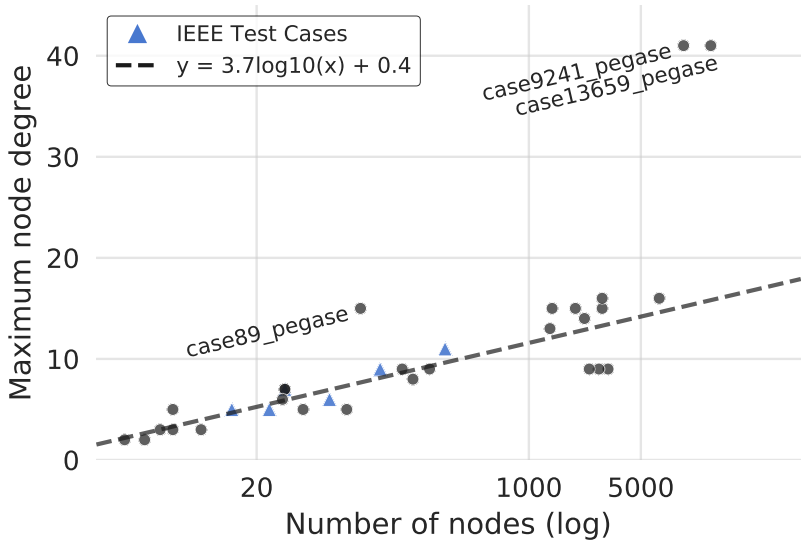


Figure 6.2: Semilog plot of maximum node degree vs. number of nodes. The dashed least-squares fit line excludes the labeled outliers.

The average mean node degree for our NESTA sample is 2.70, with a standard deviation of 0.53. Table 6.1 contains the data, along with a "rugplot" illustration of the distribution of values. Note the tight groupings about 2.4 (RTE and Polish networks) and 2.7 (PEGASE and IEEE networks, highlighted pink and blue respectively). Highly-meshed transmission networks like the 162-bus IEEE dynamic test case may naturally have high mean degree, but the PEGASE 89-bus network is clearly unusual. On the other end of the distribution, case189_edin appears to have long paths and low meshing.

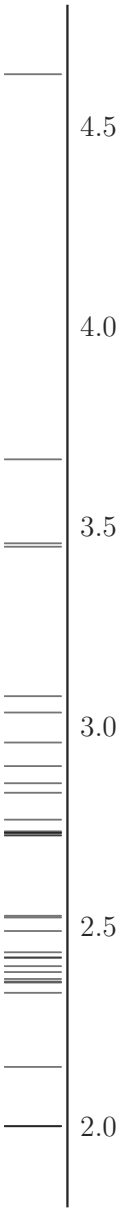
Median node degree data are less interesting. Of our NESTA sample networks, 76% have $k_{\text{med}} = 2$, while remaining values are 3. Unlike mean degree, k_{med} is not sensitive to whether generator step-up transformers are modeled, at least for NESTA networks. Dense networks like case_162_ieee_dtc (which has been described as "quite robust" [101]) may have a median node degree of 3, but this is unusual for networks larger than a few hundred nodes. The largest NESTA network with $k_{\text{med}} = 3$ is the IEEE 300-bus test case.

6.3.2 Degree assortativity

Figure 6.3 suggests that transmission systems tend to be slightly disassortative. Over half of the NESTA sample networks have $|r| < 0.2$, and one standard deviation within the mean corresponds to roughly $(-0.3, 0.15)$. The two largest PEGASE networks are highly assortative, as first shown in [11]. The case13659_pegase variant is less assortative due

Table 6.1: Mean node degree values for representative NESTA networks.

	Mean node degree
case89_pegase	4.629
case6_ww	3.667
case162_ieee_dtc	3.457
case29_edin	3.448
case9241_pegase	3.075
case118_ieee	3.034
case73_ieee_rts	2.959
case240_wecc	2.900
case14_ieee	2.857
case24_ieee_rts	2.833
case2869_pegase	2.766
case57_ieee	2.737
case30_ieee	2.733
case30_as	2.733
case30_fsr	2.733
case13659_pegase	2.727
case300_ieee	2.727
case1354_pegase	2.526
case2224_edin	2.522
case6515_rte	2.488
case1951_rte	2.435
case2383wp_mp	2.422
case2868_rte	2.421
case5_pjm	2.400
case2736sp_mp	2.385
case1397sp_eir	2.368
case3120sp_mp	2.362
case39_epri	2.359
case6_c	2.333
case189_edin	2.148
case4_gs	2.000
case9_wscc	2.000
case3_lmbd	2.000



primarily to the thousands of generator step-up transformers it includes, which are omitted in case9241_pegase. On the other end of the spectrum, case9_wscc is highly disassortative. This tiny network consists of a ring of six nodes, three of which have leaf nodes attached. The unusual disassortativity arises from the fact that no edge joins nodes with matching degree. Although highly disassortative networks tend to be tiny, and highly assortative networks are typically in the "medium" range, there is no evidence of a size-related trend

overall.

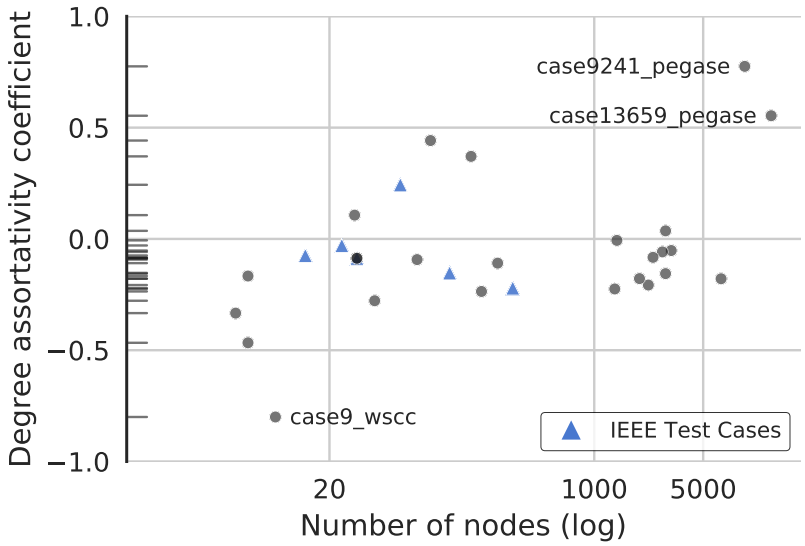


Figure 6.3: Semilog plot of degree assortativity vs. network size.

6.3.3 Rich-club effect

Table 6.2 lists all groups of three or more nodes in NESTA networks that are connected by at least 80% of their potential edges (we omit "clubs" consisting of just two nodes; see Section 6.2.3). The first row reads: "At least 80% of all possible edges exist between the 28 nodes in case9241 with degree greater than 25." The first two rows describe rich clubs first identified in [11], and the next two rows also concern PEGASE networks. The brevity of Table 6.2 is perhaps more significant than its contents. In every other case where $\phi(k)$ reaches 0.8 for a NESTA network, the "rich club" consists of just two nodes. For a dozen networks, there is no k for which $\phi(k)$ reaches 0.8. Mean node degree makes one PEGASE network stand out, degree assortativity draws attention to two, and three PEGASE networks have unusually high maximum node degree. The rich-club coefficient is the first metric that separates all four PEGASE networks from the rest of NESTA.[2]

6.3.4 Cliques

The vast majority of power grid maximal cliques contain just two nodes. Of the 124,211 maximal cliques in our representative NESTA networks, the mean size is 2.085. This

[2]Table 6.2 also lists case3_lmbd, but this "rich club" contains the entire network: three nodes in a ring.

Table 6.2: All NESTA rich clubs with more than 80% of potential edges and at least 3 nodes.

	Nodes involved	Degree
case9241_pegase	28	>25
case13659_pegase	19	>29
case89_pegase	11	>11
case2869_pegase	5	>13
case3_lmbd	3	>0

empirical observation manifests in a linear relationship between the number of maximal cliques in a power system and the number of nodes. While the number of maximal cliques grows linearly with the number of nodes, the size of the largest clique (or "maximum clique") does not. Figure 6.4 plots maximum clique size against network size for our sample networks. As with the rich-club coefficient, this metric makes the four PEGASE networks stand out. Setting aside these networks, the least-squares fit would effectively be a horizontal line. Although PEGASE networks have unusually large cliques, roughly 99% consist of just two or three nodes.

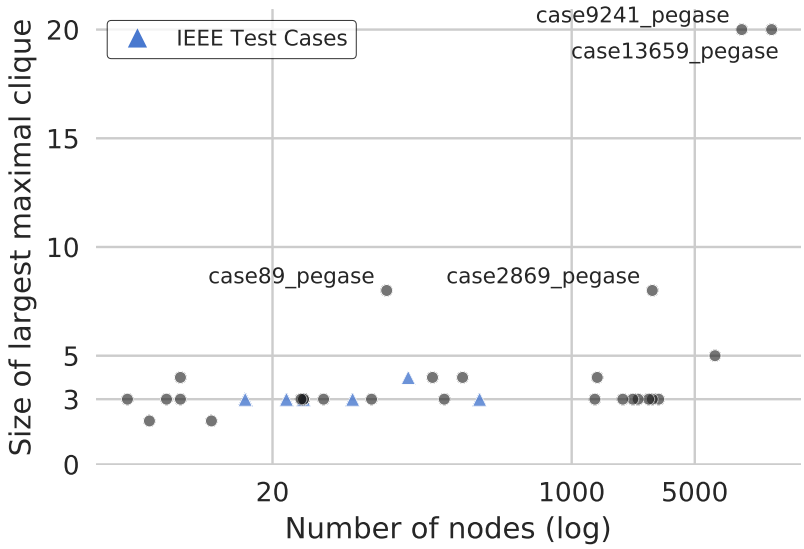


Figure 6.4: Maximum clique size versus network size.

6.3.5 Adjacency matrix spectrum

Figure 6.5 illustrates adjacency spectral radius versus number of nodes. As with the rich-club coefficient, this metric casts the four PEGASE networks as outliers. All other networks lie close to the trendline, which increases gradually with the log of network size. While the isolation of all PEGASE networks is compelling, previously-mentioned metrics provide similar information with significantly less computation.

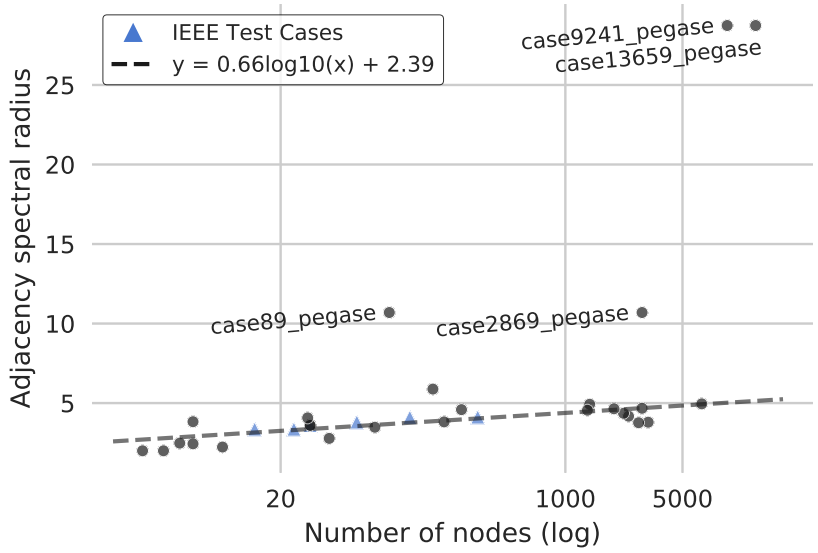


Figure 6.5: Semilog plot of largest adjacency matrix eigenvalue versus number of nodes. The dashed fit line excludes the labeled outliers.

6.4 Conclusions

A number of topological graph metrics were applied to 33 networks in NESTA, and several structural anomalies were identified. Our metrics drew attention to PEGASE networks: each metric highlighted at least one of these networks, but three metrics (rich-club coefficient, maximum clique size, and adjacency matrix spectral radius) separated all four from the rest of NESTA. Our results indicate that each PEGASE network contains a relatively small but highly dense subnetwork, which appears to be the result of Kron network reduction or a similar technique. A combination of degree distribution, degree assortativity, and rich-club coefficient metrics is effective for identifying this structure within a large graph, yet also computationally inexpensive. These metrics may be used to quickly scan new grid models for potential modeling or data issues. We also touched on the connection between cliques and SDP OPF performance. This connection is the subject of the next chapter.

CHAPTER 7

System structure and the semidefinite relaxation of the optimal power flow problem

7.1 Introduction

Advances in semidefinite programming have drawn the attention of the power systems community, specifically in connection with optimal power flow (OPF). Though not exact for all OPF problems [102, 34], the semidefinite relaxation finds global optima for many test cases [103, 104]. While extensive discussion of SDP OPF properties is beyond the scope of this chapter, we will consider one key performance aspect: SDP solvers work best with semidefinite constraint matrices no larger than a few dozen elements to a side. This makes the decomposition technique outlined in [16] essential for solving larger SDP OPF problems [105]. The positive semidefinite matrix completion theorem in [106] governs valid decompositions [94]. Let there be an undirected chordal graph with associated incomplete matrix G. Then G can be completed to a positive semidefinite matrix if and only if all submatrices associated with the graph's maximal cliques are positive semidefinite.[1] Thus, decomposition involves 1) forming a chordal extension of the power grid graph (see Section 6.2.5), 2) identifying chordal extension maximal cliques (see Section 6.2.4), and 3) replacing the semidefinite constraint matrix with a set of constraints corresponding to maximal clique submatrices.

A single node may belong to many maximal cliques. Each of these cliques has its own copy of that node's two variables (real and imaginary complex voltage phasor components in the OPF setting), but each physical parameter must ultimately have one value. Thus, maximal clique decomposition introduces a number of linking constraints that varies with the amount of overlap between cliques. Since SDP solvers like Mosek apply primal-dual

[1] In the SDP setting, completing G refers to the process of assigning values to otherwise unconstrained elements such that the resulting matrix is positive semidefinite. The SDP relaxation is feasible if such a completion exists.

methods, a primal linking constraint corresponds to a dual variable, and changes in the number of constraints and variables have similar impacts on solver performance [16]. For this reason, we will use "approximate SDP OPF problem size" to denote the sum of all scalar variables and linking constraints. If merging two cliques eliminates enough linking constraints to offset the resulting increase in submatrix size, the sum of variables and linking constraints is reduced overall, suggesting an improvement in SDP solver performance according to the approximate problem size heuristic. For this reason, [16] proposed a greedy clique merging algorithm that repeatedly combines the pair of cliques whose merger yields the greatest reduction in the sum of variables and linking constraints. Suppose the algorithm stops when a specified number of submatrices, designated L, is reached. As shown in [16], SDP solver time gradually decreases as L shrinks (i.e. as more cliques are merged). If L becomes too small, however, the submatrices grow large enough to outweigh the benefits of linking constraint elimination. In this chapter we consider relationships between chordal graph extensions, clique characteristics, clique merge algorithms, and SDP OPF performance. We first compare sparsity-preserving chordal extensions to more dense extensions. Next, we compare greedy clique merge with a modified merge algorithm that minimizes treewidth, and consider whether cliques consisting of just two nodes should be merged.

7.2 Problem setting

7.2.1 Chordal extensions and sparsity

As mentioned in Section 6.2.5, a graph is chordal if every cycle (i.e. loop) of at least four vertices has a chord, which is defined as an edge between two non-adjacent vertices in the cycle. Any graph may be extended to form a chordal graph via the addition of chords. A chordal graph extension may also be referred to as a triangulation, since every cycle of four or more nodes contains a chord-induced triangle. Chordality is also linked to graph elimination: a graph is chordal if and only if it has a perfect elimination ordering [107]. This property means that the maximal cliques of a chordal graph may be found in linear time. Given a perfect elimination ordering and its first vertex v, a maximal clique containing v may be formed by adding the neighbors of v that succeed it in the ordering. All other maximal cliques may be found similarly by proceeding through the elimination ordering. This connection between chordality and graph elimination provides a convenient means of obtaining chordal extensions: perform graph elimination using any vertex ordering, and the set of edges added due to fill-in is sufficient to render the graph chordal. The number

of filled-in edges (and therefore the density of the resulting chordal extension) depends greatly on the vertex ordering. One typically desires to promote sparsity by minimizing fill-in. Truly minimizing fill-in is an NP-hard problem, but effective vertex orderings have been proposed by Tinney and others [108, 109, 110]. One such ordering is included implicitly in implementations of sparse Cholesky factorization. The following process may be used to obtain a sparsity-preserving chordal graph extension via Cholesky factorization. Let A be the adjacency matrix of a non-chordal graph, and D be a diagonal matrix such that $D_{i,i}$ is equal to the sum of the i-th row of A. Now define a Hermitian matrix $W = A + D$. Then Cholesky factorization of W yields an adjacency matrix corresponding to a chordal extension of the non-chordal graph.

The maximal cliques of any chordal graph extension give rise to a valid semidefinite constraint decomposition. Maximal cliques may also be merged to obtain other constraint decompositions. We turn next to a description of clique merge algorithms.

7.2.2 Clique merge

Once a chordal extension has been chosen and its maximal cliques identified, cliques may be merged to reduce the number of semidefinite constraints and eliminate linking constraints. The clique graph provides a useful means of understanding clique merging. Each node in this graph represents a maximal clique. Each edge lies between a pair of cliques that share one or more nodes, and edge weight is equal to the number of nodes shared. Merging two cliques corresponds to eliminating an edge of the clique graph, replacing the endpoint cliques with one larger group of nodes. This process may be repeated until all nodes are eliminated, at which point there is a single group consisting of all nodes. To eliminate as many linking constraints as possible, it is sufficient to consider only clique graph edges belonging to its maximum-weight spanning tree. The clique graph maximum-weight spanning tree is obtained efficiently using Prim's algorithm [111].

At each step, the greedy clique merge algorithm eliminates an edge of the clique graph, merging its endpoints into a single group of nodes. To choose which edge to eliminate, the algorithm estimates the change in approximate problem size that would result from eliminating each clique graph edge. By its nature, greedy clique merge will only eliminate clique graph edges belonging to the maximum-weight spanning tree described above. Denote this tree $T = \{\mathbf{V}, \mathbf{E}_T\}$, where $\mathbf{V}$ represents the clique graph vertices, and $\mathbf{E}_T$ represents the set of edges. If a node $\mathbf{v}_i \in \mathbf{V}$ represents a chordal extension maximal clique consisting of $|\mathbf{v}_i|$ buses (each of which has real and imaginary voltage phasor components), then the corresponding OPF semidefinite constraint matrix is square with $2|\mathbf{v}_i| \times 2|\mathbf{v}_i|$ ele-

ments, and $|\mathbf{v}_i|(2|\mathbf{v}_i|+1)$ unique optimization variables in its upper triangle. Similarly, an edge $e_i \in \mathbf{E}_T$ with weight w_{e_i} represents a clique overlap of w_{e_i} buses, which introduces $w_{e_i}(2w_{e_i}+1)$ linking constraints due to duplication of variables. The approximate SDP OPF problem size corresponding to a clique graph spanning tree T is the sum of all scalar variables and linking constraints:

$$v(T) = \sum_{\mathbf{v}_i \in \mathbf{V}_T} |\mathbf{v}_i|(2|\mathbf{v}_i|+1) + \sum_{e_i \in \mathbf{E}_T} w_{e_i}(2w_{e_i}+1). \tag{7.1}$$

With the clique graph and approximate problem size defined, implementation of the greedy clique merge algorithm is straightforward. At each step, the algorithm compares $v(T)$ with $v(T_{-e_i})$ for each edge $e_i \in \mathbf{E}_T$, where T_{-e_i} is obtained by merging the endpoint cliques of e_i. The merge corresponding to the greatest reduction in approximate problem size is implemented, and T is updated. Merging may be stopped once $|\mathbf{V}|$ reaches a predefined threshold, when subsequent merges begin increasing $v(T)$, or according to another termination criterion.

7.2.3 Treewidth and 2-cliques

In addition to sparsity, treewidth (defined here as the size of the largest clique or group of merged cliques) and the modeling of 2-cliques (cliques consisting of two nodes) are also important concerns. For a given number of clique merges, lower treewidth means a greater number of smaller node groups. A greedy clique merge algorithm may be set up to exhaust all merges that do not increase treewidth before forming any larger groups. We hypothesize that this prioritization of small semidefinite constraints will lead to improved SDP OPF solver performance. The other consideration is whether to leave 2-cliques out of clique merging altogether. It is well-known that a size 2-by-2 semidefinite constraint may be expressed using an equivalent second-order cone constraint. All semidefinite constraints corresponding to 2-cliques may therefore be replaced by conic constraints, which should improve optimization performance.

Four greedy merge algorithm variants were implemented to determine the impact of minimizing treewidth and setting aside 2-cliques during clique merge. The first is standard greedy clique merge. The second is greedy clique merge with all 2-cliques are excluded (and modeled using conic rather than semidefinite constraints). The third algorithm avoids increasing treewidth until necessary, merging only smaller groups until none remain. The fourth algorithm minimizes treewidth and sets aside 2-cliques (allowing them to be modeled with conic constraints). Results of applying these algorithms to the 300-bus IEEE test

network are shown in Section 7.3.3.

7.3 Results

7.3.1 Chordal extension sparsity and SDP OPF performance

The choice of chordal extension sets the stage for clique merge and SDP OPF performance. Perhaps the most important difference between various chordal extensions is sparsity. A chordal extension may be obtained by performing graph elimination with any vertex ordering, and sparsity-preserving "approximate minimum degree" orderings like Tinney-2 [110] lead to sparse chordal extensions. The Tinney-2 approach begins by identifying the graph node with lowest valence (fewest neighbors), where any tie is broken randomly. This node is then eliminated, and the adjacency matrix is updated with the fill-in elements (if any) introduced by elimination of the first node. The node with fewest neighbors in the updated adjacency matrix is eliminated next, and the process is repeated until the ordering includes all vertices and graph elimination is complete. To assess the impact of chordal extension sparsity on clique merge behavior and SDP OPF performance, we began with a Tinney-2 vertex ordering and gradually altered it to obtain a family of orderings with gradually increasing fill-in. Each ordering was generated as follows. Let the Tinney-2 ordering be split into three parts: the part which does not lead to any fill-in (which is at the beginning, and is left alone), the next n elements, and all remaining elements. Reverse the order of the remaining elements to obtain a new ordering. Then increase n and repeat. This gradual deviation from the initial sparsity-preserving ordering leads to a corresponding increase in chordal extension density. A family of vertex orderings was generated in this manner for the IEEE 300-bus test case, and the SDP OPF was solved for each ordering. Two trials were considered: one where no clique merging was performed, and another where greedy clique merge was used to minimize approximate problem size. Figure 7.1 illustrates the relationship between optimization time and chordal extension density for both cases, on a log-log scale. As might be expected, chordal extension sparsity leads to reduced optimization time. More interesting is the tendency of greedy clique merge to perform better for sparser extensions, and worse for dense extensions. The algorithm's ability to achieve a reduction in SDP OPF solver time by minimizing approximate problem size depends on the sparsity of the chordal extension. While these results do not determine whether better chordal extensions are possible, they do suggest that a sparsity-preserving chordal graph extension is an appropriate starting point for clique merging.

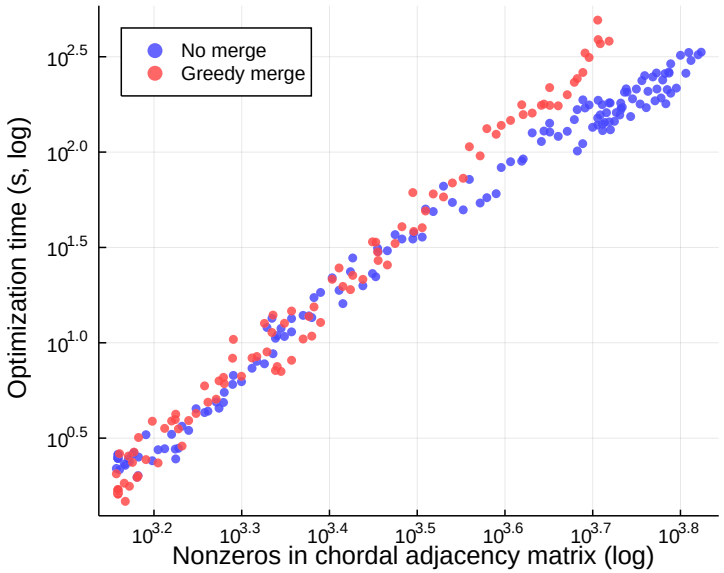


Figure 7.1: SDP OPF optimization time versus number of nonzero chordal adjacency matrix elements for permuted Tinney-2 orderings, on a log-log scale. The underlying network graph is from the IEEE 300-bus system.

7.3.2 Clique merge behavior for NESTA networks

Behavior of the greedy clique merge algorithm depends on system size and structure, as well as sparsity of the chordal graph extension. To illustrate greedy clique merge behavior in the SDP OPF setting for a wide range of networks, we applied the method to several NESTA [57] networks. Because greedy clique merge tends to perform better for sparse chordal extensions, we used sparse Cholesky factorization to obtain all chordal extensions. Throughout algorithm execution, we tracked the number of variables and linking constraints, the sizes of the largest and smallest groups of nodes, and Mosek SDP OPF solver times. Detailed results are shown for the IEEE 300-bus test case in Figure 7.2, which tracks a few key features at each clique merge step. The size of the largest group (top solid greed line) is initially equal to the maximum clique size. This value rises sharply at first, then remains flat until the last few merges.[2] The size of the smallest group (bottom green line) remains 2 until the last remaining clique is merged with another group during merge 212. Approximate problem size decreases initially, as the greedy algorithm eliminates linking constraints. It reaches a minimum (indicated by the "X" on the Variables curve) after roughly half of all merges have occurred. Beyond this point, approximate problem size as expressed in (7.1) increases due to the formation of larger groups of nodes. Despite

[2]Many cliques must be merged into medium-sized groups of nodes before it makes sense to merge these larger clusters together.

this fact, solver times continue to drop further, until nearly 200 merges have occurred. In this case, solution time tends to be minimized close to the point at which the last clique is merged into another group (indicated by the jump in smallest group size).

Other NESTA networks we tested behave similarly to the IEEE 300-bus case. Three quantities of interest were tracked while applying greedy clique merge and solving the SDP OPF for several networks: fraction of merges until $v(T)$ is minimized, fraction of merges until the last clique is merged into another group, and fraction of merges until solution time is minimized. This latter value was estimated by performing 50–250 Mosek trials[3] to obtain a spread of solver time data for each merge index, discarding points in each spread more than two standard deviations from the mean, and inally averaging the remaining trials. This computationally intensive process yielded stable average optimization times, but rendered it impractical to consider networks larger than a few thousand nodes. Figure 7.3 illustrates results. Approximate problem size $v(T)$ is typically minimized after around 50% of merges, the last unmerged clique is usually merged after roughly 75% of merges have taken place. Note that the gap between these points is wider for larger systems. Solver time tends to be minimized well after $v(T)$ is minimized, closer to where the last maximal clique is merged. A stopping condition based only on minimizing approximate SDP OPF problem size will therefore terminate the greedy clique merge algorithm too soon. This calls for an improved stopping condition for greedy clique merge, or perhaps a different merge algorithm altogether.

7.3.3 Comparison of four clique merge algorithm variants

Given the high impact of treewidth on computational performance, and the ability to model 2-cliques using conic constraints (rather than semidefinite constraints), it seems plausible that clique merge algorithms which minimize treewidth and avoid merging 2-cliques should out-perform standard greedy clique merge. To test this hypothesis, we applied the four algorithms described in Section 7.2.3 to the IEEE 300-bus test network. Figure 7.4 illustrates treewidth and optimization time versus merge index. While greedy clique merge seeks to minimize $v(T)$ (Equation (7.1)) without regard for treewidth, the min-treewidth algorithm does not perform any merge that would increase the largest group size until it become necessary. The top plot shows that min-treewidth clique merge keeps the treewidth at 11 until around merge 120 if 2-cliques are kept aside, and around merge 180 otherwise. By contrast, greedy merge increases treewidth right away. Min-treewidth has a slight solver time advantage, which becomes more pronounced after many merges have been performed. (Solver

[3] Smaller networks required more trials due to increased variance relative to mean solver time.

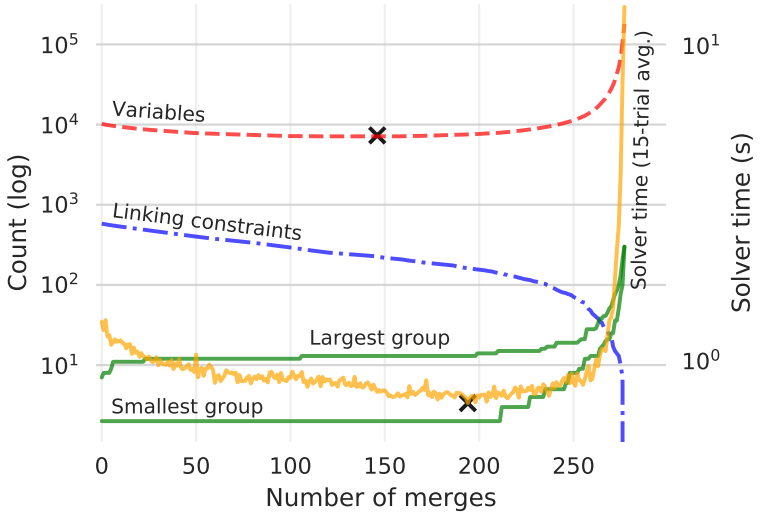


Figure 7.2: Semilog plot illustrating clique merge behavior for the IEEE 300-bus test case. Solver time is plotted in yellow against the right axis; all other quantities are plotted against the left axis. Minimum variable count and minimum solution time are each indicated with "X".

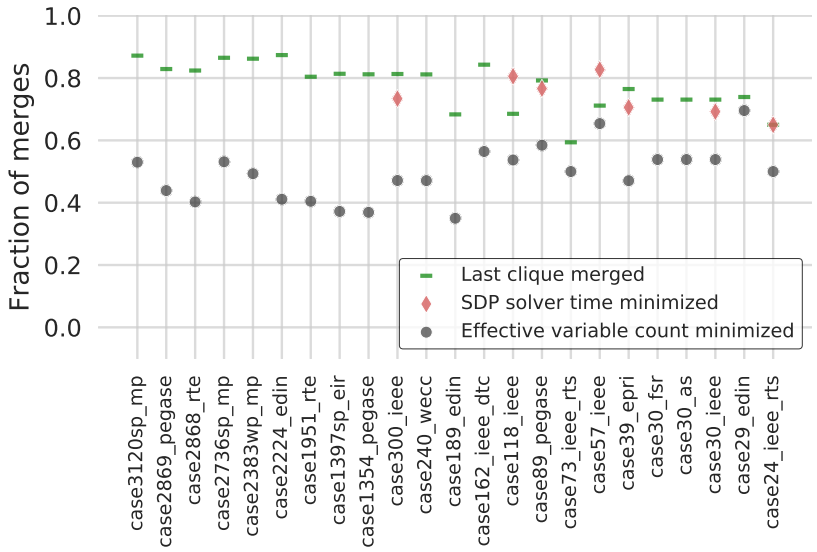


Figure 7.3: Merges required to reach important points of clique merge algorithm execution.

time has increased substantially by this point, and anyone seeking to minimize solver time would have already stopped merging. The gap at higher merge counts is therefore irrelevant.) When 2-cliques are left out of clique merge (to be modeled as conic constraints), the increases in treewidth and solver time occur after fewer merges, since only groups with more than 2 nodes may be merged. Aside from this, there is no apparent benefit to replacing

semidefinite constraints for 2-cliques with second-order cone constraints.

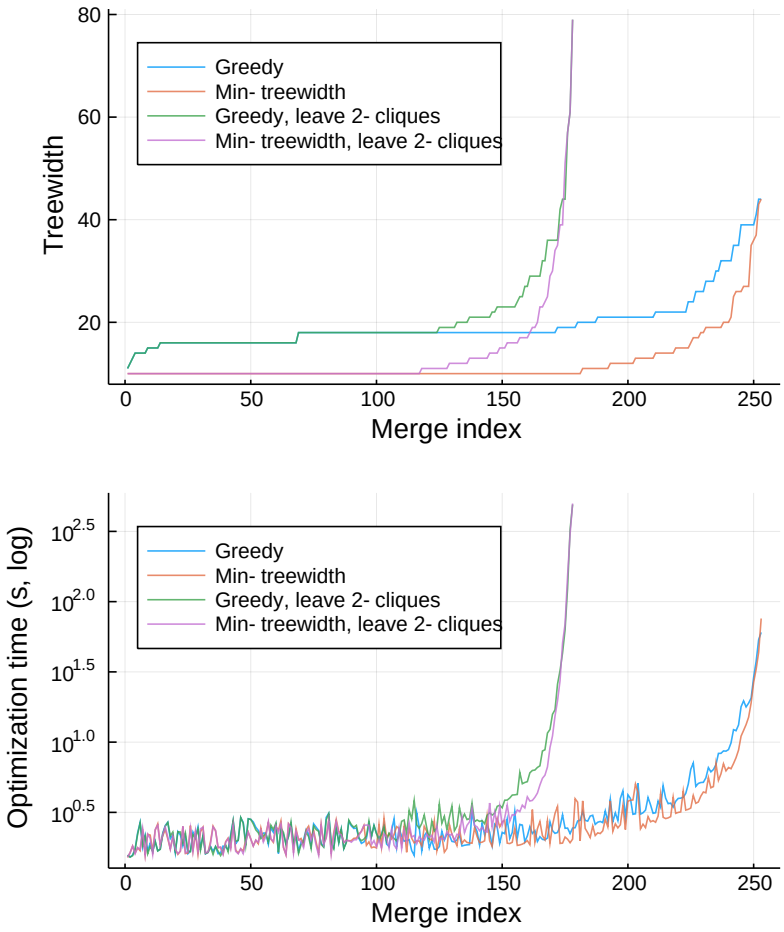


Figure 7.4: Treewidth (top) and SDP OPF optimization time (bottom) versus merge index for four clique merge algorithms. The underlying network graph comes from the IEEE 300-bus system.

7.4 Conclusions

This chapter explored the connection between chordal graph extensions, clique graph structure, clique merge algorithms, and SDP OPF optimization performance. It was shown that sparsity is desirable in chordal extensions not just because it improves numerical performance, but also because it allows greedy clique merge to better reduce optimization time. It was then confirmed that greedy clique merge, the state of the art merging algorithm in the literature, performs better than either the full semidefinite constraint or the fully decomposed set of semidefinite constraints. It does not, however, identify the decomposition

that minimizes solver time (at least not when minimization of approximate problem size is used as the stopping criterion). This calls for improved merging algorithms, and three were proposed and tested. The min-treewidth algorithm avoids increasing the size of the largest group until necessary. Results for this algorithm were as good as, or even slightly better than, those of greedy clique merge. The other two algorithms implemented greedy merge and min-treewidth merge, but kept 2-cliques aside to allow them to be modeled with second-order cone constraints. These algorithm variants did not lead to improved solver performance in our test case. One key takeaway is that SDP OPF solver performance varies with numerous system parameters, chordal extension properties, and clique composition. With improved merge algorithms and stopping conditions, further performance improvements are possible, though it remains unclear how to improve upon greedy clique merge consistently. There may also be a better approach for generating and evaluating semidefinite constraint decompositions than sequentially merging groups of nodes.

CHAPTER 8

Renewable fluctuation and load-tap-changing transformer operation minimization

8.1 Introduction

Load Tap Changing (LTC) transformers[1] are prevalent throughout subtransmission and distribution networks. As generation and demand fluctuate, the LTC keeps its terminal voltage within an acceptable range by altering its effective turns ratio [112]. Grid-scale renewables at the subtransmission level, with their significant active power fluctuations, can make this task more difficult for downstream distribution-level LTCs (see Fig. 8.1). Recent literature contains a variety of solutions to this problem. One popular proposal is to have grid-scale renewables perform voltage regulation. While this achieves the goal of reducing strain on downstream LTCs, it also increases upstream power flows, thereby straining subtransmission-level LTCs.[2] In this chapter we summarize recent work on LTC tap reduction in networks with grid-scale renewables, showing that many proposed solutions merely divert the burden of renewable variation from one transformer to another. We then discuss two popular LTC models. Finally, we use publicly available wind and solar generation data and a simple test network to illustrate the trade-off between subtransmission and distribution LTC tapping for four combinations of LTC model and renewable generation type.

8.2 LTC tap reduction review

LTC limitations are numerous and well-known. Tap-changer lifespan is estimated to be roughly 50,000 taps [115] (though recently-developed vacuum-type devices may last be-

[1] Other acronyms for these devices include "Tap-Changing-Under-Load" (TCUL), "On-Load-Tap-Changing" (OLTC), and "Under-Load-Tap-Changing" (ULTC). The key characteristic is online operation: transformers whose turns ratios may only be adjusted offline are excluded.

[2] Though we restrict our attention to this side-effect, others exist [113, 114].

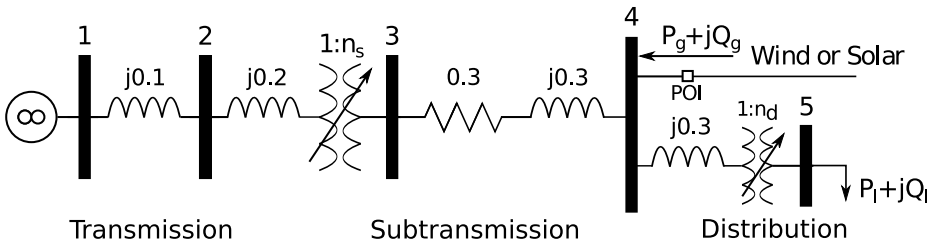


Figure 8.1: Test network.

yond a million [116]). Discrete behavior presents a second limitation: tapping provides a discontinuous response to a continuous disturbance. Deadbands and time delays work to prevent excessive or redundant tapping, but one must be sure to leave enough sensitivity for the LTC to do its job. A third limitation is tap saturation, which occurs when the transformer reaches the end of its tapping range. LTCs typically regulate $\pm$5-10% [117, 20, 19], which may be insufficient for extreme fluctuations in generation and load. Communication and control limitations comprise a fourth category of limitations. Existing LTCs are rarely equipped to send and receive signals, so remote control of setpoints and deadbands requires additional equipment. Finally, there can be undesirable interactions between voltage regulation devices in a network. These include hunting [118, 114, 119] and voltage regulator runaway [20]. Of the limitations listed, we focus primarily on limited tap-changer lifespan, which tends to be most costly. LTC failure is responsible for a significant portion of grid interruptions, and when a power transformer fails, its tap changer is the second-most-likely culprit [120].[3] A broken transformer requires substantial time and money to repair or replace [20]. Voltage control schemes must be mindful of each tap's non-negligible cost, especially in networks with significant renewables.

There are two approaches for responding to increased voltage variation due to renewables. One family of methods views the LTC as a controllable device, with inputs including voltage setpoints, deadbands, and time delays. Setpoint adjustment is a simple and field-tested approach belonging to this category. Allowing the setpoint to vary with renewable generation can reduce the number of tap operations, albeit with negative repercussions on the voltage profile [17, 18]. Deadband tuning can also extend LTC lifespan: a wider deadband means fewer tap operations (but also looser voltage regulation). Temporarily widening a deadband to coordinate with downstream capacitor banks can reduce tapping without sacrificing voltage regulation [19]. Scheduling methods take LTCs offline and plan tap positions in advance. The schedule is typically derived by minimizing voltage deviations subject to a variety of constraints. As part of the optimization, the number of LTC

[3] According to [120], tap changer failure is responsible for 27% of 20-100 MVA transformer failures and 13% of 100-400 MVA failures. As a cause of failure, the tap changer is second only to "general aging."

operations may be minimized [20], capped [21, 22, 23], or disregarded [121, 122, 123]. Scheduling is a reliable way to prevent undesirable interactions and tap saturation, but one must ensure sufficient non-scheduled (closed-loop) voltage control capability remains to respond to forecast inaccuracy. Ultimately, the family of LTC control methods just described presumes fine-grained controllability of setpoints, deadbands, tap positions, or time delays. This typically requires expensive modification of existing LTCs. Coordinated control strategies additionally assume accurate forecast data and low communication latency.[4] These assumptions may grow increasingly valid as devices and forecasting methods become more sophisticated, but such changes are many years off for typical subtransmission networks. Devices as costly as LTCs are replaced only when they fail, and many substations have on-site replacements that must be used up before innovative transformers can take over. In the meantime, voltage control methods that work with existing LTC behavior are preferable.

The second approach for alleviating LTC strain in networks with substantial renewable generation is centered around voltage regulation at renewable nodes. Instead of controlling LTCs directly, one can employ voltage control at renewable nodes to influence (and hopefully reduce) LTC tapping. This requires no LTC modification, and allows renewable energy sources to offset their burden. One might think strict renewable voltage control is a perfect solution for renewable variability. If one considers only downstream effects, this is true: simply having each wind farm regulate its power factor to 0.95 leading can significantly reduce distribution LTC tap operations [124]. Unfortunately, renewable voltage regulation induces reactive power flows which can increase strain on upstream subtransmission LTCs [114]. This trade-off was previously described in [125]. Renewable voltage regulation must balance the effects of variability between downstream and upstream transformers, determining how many tap operations are required of each.

8.3 LTC behavior and modeling

Typical LTC transformer operation is described in [19]. The device adjusts its tap position within a $\pm 10\%$ range, most often via 16 up taps and 16 down taps (in the United States, at least). Accordingly, each tap adjusts the transformer turns ratio (and hence the secondary side open-circuit voltage) by 5/8%. A voltage deadband is used to prevent excessive or redundant tapping. Most utilities use a deadband that is symmetric about the setpoint and 2-4 taps (1.25-2.5%) wide [19]. A time delay further reduces unnecessary tapping. The delay may be definite (independent of voltage deviation size) [126, 127], or it may exhibit

[4] See [18] for a discussion of latency's effects on voltage control schemes.

an inverse characteristic (where a larger voltage deviation results in a shorter delay) [128].

As with any device, there are several mathematical models one might use to represent an LTC transformer, depending on the application. The remainder of this section presents two common models we use later on. One has a definite time delay, and the other incorporates an inverse delay. We denote these models DLTC and ILTC, respectively.

The DLTC model relies on a simple counter (definite time delay) to determine when to tap. It is used in [117] and [129], and has been around for decades [130].

$$d(t) = \begin{cases} 1 & \text{if } \Delta V(t) > DB \\ -1 & \text{if } \Delta V(t) < -DB \\ 0 & \text{otherwise} \end{cases} \tag{8.1a}$$

$$c(t) = \begin{cases} c(t-\Delta t) + \Delta t & \text{if } d(t) = 1 \text{ and } c(t-\Delta t) \geq 0 \\ c(t-\Delta t) - \Delta t & \text{if } d(t) = -1 \text{ and } c(t-\Delta t) \leq 0 \\ 0 & \text{otherwise} \end{cases} \tag{8.1b}$$

$$T(t) = \begin{cases} T(t-\Delta t) + 1 & \text{if } d(t) = 1 \text{ and } c(t) > C \\ T(t-\Delta t) - 1 & \text{if } d(t) = -1 \text{ and } c(t) < -C \\ T(t-\Delta t) & \text{otherwise, and when } T \text{ at limit} \end{cases} \tag{8.1c}$$

The input signal is $\Delta V(t)$, the difference between observed and reference voltage magnitudes. The signal $d(t)$ is 0 when $\Delta V(t)$ is within the deadband, 1 above the deadband, and -1 below the deadband (8.1a). If $d(t)$ remains 1 or -1 for more than one time step Δt, the counter c increases in magnitude; otherwise, it resets to 0 (8.1b). Once the counter reaches $\pm C$, which is typically on the order of fifteen seconds to a few minutes, the tap increases or decreases once per time step (8.1c). Thus, when the terminal voltage leaves the deadband, the counter increases (decreases) until it reaches a predetermined maximum (minimum). The tap position then increases (decreases) once per time step until terminal voltage returns to the deadband or the transformer encounters a tap limit. When terminal voltage returns to the deadband, the counter is reset.

The ILTC model incorporates an inverse time delay, and was proposed in [131].

$$\frac{de(t)}{dt} = \begin{cases} \frac{1}{\tau}(\Delta V(t) - DB) & \text{if } \Delta V(t) > DB \\ \frac{1}{\tau}(\Delta V(t) + DB) & \text{if } \Delta V(t) < -DB \end{cases} \tag{8.2a}$$

$$e(t) = \begin{cases} e(t - \Delta t) + \frac{de(t)}{dt}\Delta t & \text{if } T(t) = T(t - \Delta t) \\ 0 & \text{otherwise, and when } V \text{ in deadband} \end{cases} \tag{8.2b}$$

$$T(t) = \begin{cases} T(t - \Delta t) + 1 & \text{if } e(t) > \alpha \\ T(t - \Delta t) - 1 & \text{if } e(t) < -\alpha \\ T(t - \Delta t) & \text{otherwise, and when } T \text{ at limit} \end{cases} \tag{8.2c}$$

The derivative $\frac{de(t)}{dt}$ varies directly with the gap between terminal voltage $\Delta V(t)$ and deadband DB, and inversely with time constant τ (8.2a). The "tap error" e increases or decreases based on $\frac{de(t)}{dt}$, but is reset to 0 after a tap occurs and whenever V is inside the deadband (8.2b). The transformer adjusts its tap position T when the magnitude of $e(t)$ exceeds that of one tap step α (8.2c). Thus, when input voltage leaves the deadband, tap error $e(t)$ increases (or decreases) with time constant τ, until its magnitude reaches one tap step. Then the tap position T increases (or decreases), and e is reset. A larger ΔV causes e to grow faster, which induces tapping sooner.

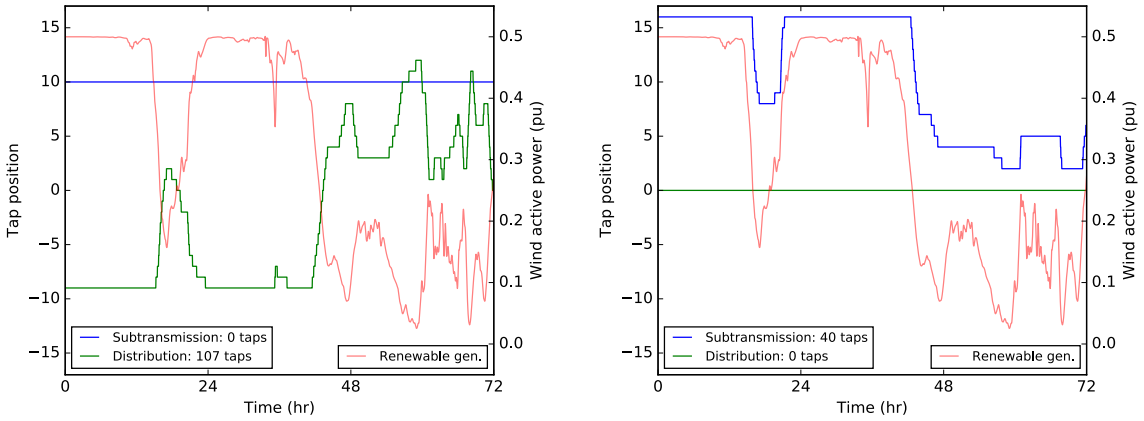
The two models share most parameters (voltage setpoints, deadbands, and tap ranges and sizes). Only the timing parameters are unique. If C and τ are tuned for a particular input signal, the DLTC and ILTC models will tap with roughly the same frequency.

8.4 Case study

To illustrate the trade-off between subtransmission and distribution LTC tapping, we have developed a case study that uses a simple test network, the two LTC models presented in Section 8.3, and renewable generation data from the National Renewable Energy Laboratory (NREL) [61, 132].

8.4.1 Test network

Fig. 8.1 depicts a radial test network that is suitable for exploring the impact of renewable generation variability on the tapping of higher- and lower-voltage transformers. A large upstream system is represented by its Thévenin equivalent behind node 2. A subtransmission LTC transformer regulates voltage at node 3. Renewable generation is connected to



(a) No renewable voltage regulation.

(b) Full renewable voltage regulation.

Figure 8.2: DLTC tap trajectories for the two extreme cases.

node 4, and we assume the maximum active power injection here is 0.5 pu. A distribution LTC lies downstream of node 4. It serves a constant load of 0.2 + j0.1 pu at node 5. Line admittances are set to values that would be realistic before the introduction of renewable generation.

Suppose there is no voltage regulation at node 4. As active power injection at the point of interconnection (POI) of the renewable generation varies, so does the fraction of load that must be supplied by the upstream power system. This leads to varying voltage magnitudes at the POI and load nodes. The distribution LTC responds by tapping to keep the load voltage within acceptable limits. Now suppose the renewable source tightly regulates voltage at node 4. As its active power output varies, the renewable generator will produce or absorb reactive power to keep its terminal voltage fixed. This variation in active and reactive power flow through the upstream network results in voltage variations which the subtransmission LTC must compensate. Lax regulation of the POI voltage sends the burden of renewable variation downstream to the distribution transformer; tight POI voltage regulation sends it upstream to the subtransmission transformer. A trade-off curve of intermediate scenarios lies between these extremes, as we will soon illustrate.

8.4.2 LTC parameters

For comparison purposes, we used both LTC models described in Section 8.3 in our case study. We selected identical tap ranges and deadbands for the two models based on typical values in the literature. Each model has 16 up taps and 16 down taps, and each step adjusts downstream voltage by 5/8%. Voltage setpoints are 1.02 pu for subtransmission and 0.96

pu for distribution,[5] and each model has a deadband of two tap steps in either direction (i.e. $\pm 1.25\%$). Timing parameters were determined via simulation, using the renewable generation data discussed in Section 8.4.3. Our goal was to choose values for C and τ that cause the DLTC and ILTC models to tap roughly the same number of times over the course of a year. We chose $C = \tau = 120s$ for the subtransmission LTCs, and $C = \tau = 240s$ for the distribution LTCs.[6] Although identical values for C and τ happened to yield similar tap frequencies with our network and renewable data, this will not always be the case. Also, matching tap frequencies do not necessarily imply matching tap behavior, as we will see in Section 8.4.5.

8.4.3 NREL renewable generation data

With load assumed to be constant, the sole source of variation in the system is renewable active power injection at node 4.[7] Because wind and solar behave quite differently, we consider both sources. The United States' National Renewable Energy Laboratory (NREL) makes its Wind Integration National Dataset (WIND) freely available [61]. WIND combines meteorological data with carefully-selected turbine power curves to simulate wind power production at over 126,000 sites in the United States across seven years. We arbitrarily chose to use wind data from site no. 16345, simulated for the year 2012.[8] Despite its impressive span, the NREL dataset is temporally limited from the perspective of LTC dynamics. Its five-minute time resolution is too low to assess LTC dynamics, which may occur at tens of seconds to a few minutes in real networks [127, 133, 126]. As a compromise, we used cubic spline interpolation between data points to obtain a one-minute-resolution time series.[9] Then we chose fairly realistic LTC time delays as described in Section 8.4.2 to ensure tapping did not occur more than once per minute. NREL publishes a solar integration dataset similar to WIND, with simulated five-minute solar power data for many locations in the United States during the year 2006 [132]. We arbitrarily selected a site in Michigan with coordinates 41.85° N, -83.55° E. As with the wind data, we used cubic spline interpolation to obtain a yearlong time series with one-minute resolution.

[5] The low setpoint of 0.96 pu keeps the distribution LTC from sitting at its +16 tap position too often, which would obscure the trade-off phenomenon of interest. In a real network, the setpoint would be closer to 1 pu, and a capacitor bank might be used to boost load voltage.

[6] The distribution LTC's longer delay allows it to wait until upstream voltage settles, preventing undesirable interactions between transformers [128].

[7] We hold load constant to separate the effects of renewable generation variability.

[8] A full year of data contains important seasonal variation [124].

[9] This technique was also used, albeit with the addition of noise, in [132].

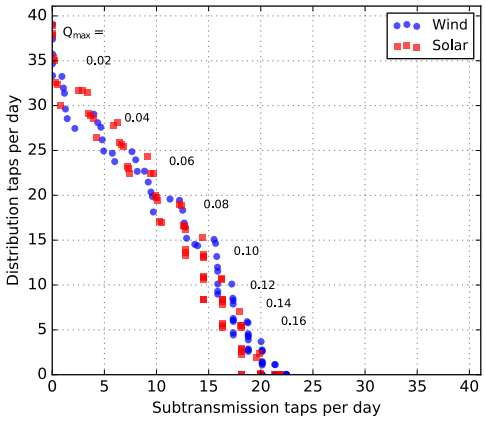
8.4.4 Tap trajectories

Fig. 8.2 shows how the DLTC transformers respond to wind variation with and without regulation of the POI voltage. The level of renewable active power output determines the voltage difference between buses 3 and 4. When output is high and constant, there is more than enough local generation for the load, and voltage at bus 4 is higher; when output is low, additional power must be drawn from the transmission network, and voltage at bus 3 is higher. Reactive power output capability at bus 4 determines which of the two voltages is the stronger regulation point. Without reactive capability at the POI, bus 4 floats relative to bus 3, and the distribution LTC must compensate (Fig. 8.2a). With unlimited POI reactive capability, bus 3 floats relative to bus 4, and the subtransmission LTC must compensate (Fig. 8.2b). There are many scenarios between these extremes, as the next section will illustrate.

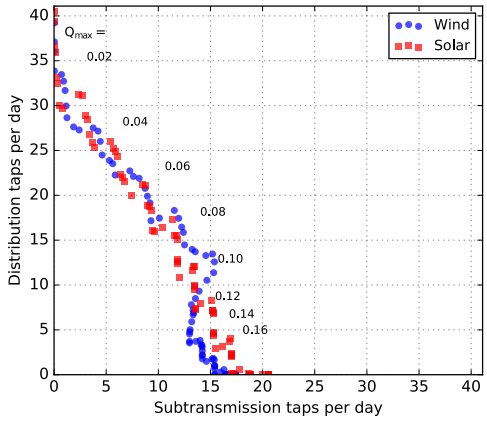
8.4.5 Trade-off curves

Reactive power capability at bus 4 splits the effects of active power fluctuations between the two LTCs in the network of Fig 8.1. If POI reactive capability is zero, voltage control is effectively turned off. As it increases, the renewable generator provides increasing voltage support at node 4. One natural way to visualize the effects of this gradual increase in voltage support is a trade-off plot. We used the following procedure to generate four trade-off series, one for each combination of LTC model (DLTC or ILTC) and NREL data type (wind or solar). 1) Let $\pm Q_{\max}$ be the reactive power limits at the renewable node (bus 4 in Fig. 8.1). In other words, let bus 4 be a voltage-controlled bus unless the necessary reactive injection is greater than $+Q_{\max}$ or less than $-Q_{\max}$. 2) Now let $Q_{\max}$ take 100 values between 0.0 and 0.25 pu. 3) For each of these limit values, perform a yearlong simulation of the Fig. 8.1 network at a one-minute resolution, with active injection at bus 4 coming from the NREL data described in Section 8.4.3. 4) Count subtransmission and distribution LTC operations for each simulation, and determine the average number of taps per day. 5) Plot these tap frequencies against each other to obtain a scatterplot series with 100 points. The four series we obtained via this procedure are plotted in Fig. 8.3a and 8.3b. We also plotted the total frequency of tap operations (subtransmission plus distribution) versus $Q_{\max}$ in Fig. 8.4a and 8.4b.

The four series in Fig. 8.3 possess the same basic shape. The slope of each is about -2, suggesting that a two-tap reduction in distribution LTC operations results in one additional subtransmission LTC operation. Fig. 8.4 indicates that regardless of LTC model or renewable generation type, total tap frequency (vertical axis) tends to decrease with increasing

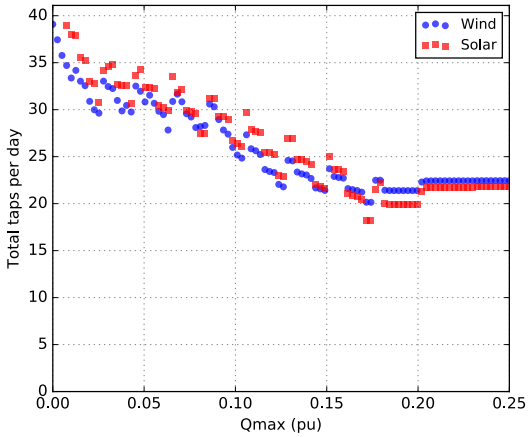


(a) Definite time delay (DLTC) model.

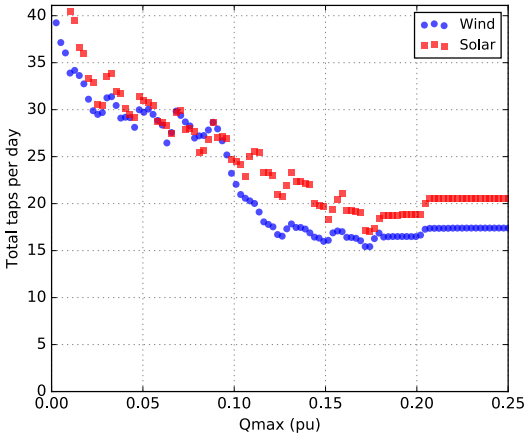


(b) Inverse time delay (ILTC) model.

Figure 8.3: Trade-off curves.



(a) Definite time delay (DLTC) model.



(b) Inverse time delay (ILTC) model.

Figure 8.4: Total number of taps as a function of renewable reactive power limit.

reactive power limit (horizontal axis). If the two transformers are equally costly to replace, renewable voltage regulation is desirable, as it leads to fewer tap operations overall.

There are a few notable differences between the four series. The two series in Fig. 8.3a are slightly further from the origin than those of Fig. 8.3b, implying the DLTCs tap slightly more often than the ILTCs overall. This is due to the balance between maximum DLTC counter value C in (8.1) and ILTC time constant τ in (8.2). To ease comparison between the two models, we attempted to match tap frequencies (see Section 8.4.2), but this is not an exact process. There are also differences between wind and solar fluctuations. For one, the latter tend to induce more tapping due to their diurnal nature. Regardless of its daytime peak, active solar injection at bus 4 drops to zero at night. Wind variation does not typically

impose such extreme variation on LTCs. Solar generation also results in slightly steeper trade-off curves, implying greater reduction in total tap operations as renewable reactive capability increases. This is apparent in Fig. 8.4a and the region to the left of $Q_{\text{max}} = 0.09$ pu in Fig. 8.4b. Between $Q_{\text{max}} = 0.09$ pu and $Q_{\text{max}} = 0.14$ pu, there is a difference between wind and solar that appears only when the ILTC model is used. The wind trade-off curve in Fig. 8.3b has positive slope in this region, and the corresponding curve in Fig. 8.4b drops rapidly. In general, increasing Q_{max} will cause the subtransmission LTC to tap more and the distribution LTC to tap less, so the trade-off curve will have negative slope ($Q_{\text{max}} = 0$ to 0.09 pu). But when the subtransmission LTC encounters a tap limit, it cannot tap more despite facing additional burden from wind variability. Both transformers will tap less, the trade-off curve will have positive slope, and the voltage profile will suffer ($Q_{\text{max}} = 0.10$ to 0.14 pu). Finally, if enough wind variability burden is shifted to the subtransmission LTC, the distribution LTC will tap rarely or not at all. Whenever the subtransmission LTC is not saturated, it will tap more to compensate, and the trade-off curve will have negative slope once more ($Q_{\text{max}} = 0.14$ to 0.18 pu). This subtle difference between DLTC and ILTC behavior for a particular choice of parameter values illustrates the complexity of the transformer tapping trade-off for even the simplest networks.

An LTC performs fewer tap operations its voltage remains within the deadband more often, but also performs fewer operations if its tap position is stuck at a limit (e.g. the LTC would continue tapping if it could, but is unable due to physical limitation). In this case, the LTC will not wear out, but only because the device is incapable of performing its voltage regulation function; system voltage will suffer as a result. The trade-off curves of Figure 8.3 do not illustrate the extent to which the LTCs maintained a desired voltage profile. We add this information to subsequent plots by introducing a voltage profile quality metric V_Q. Let $\mathbf{V}_t$ represent the vector of bus voltage magnitudes at time step t of a quasi-static simulation, and $\mathbf{V}_{\text{nom}}$ the desired voltage profile. Before simulation begins, let $V_Q = 0$. At each time step t (from 1 to the final time step t_{end}) during simulation, V_Q is increased by $\|\mathbf{V}_t - \mathbf{V}_{\text{nom}}\|$. Thus we have

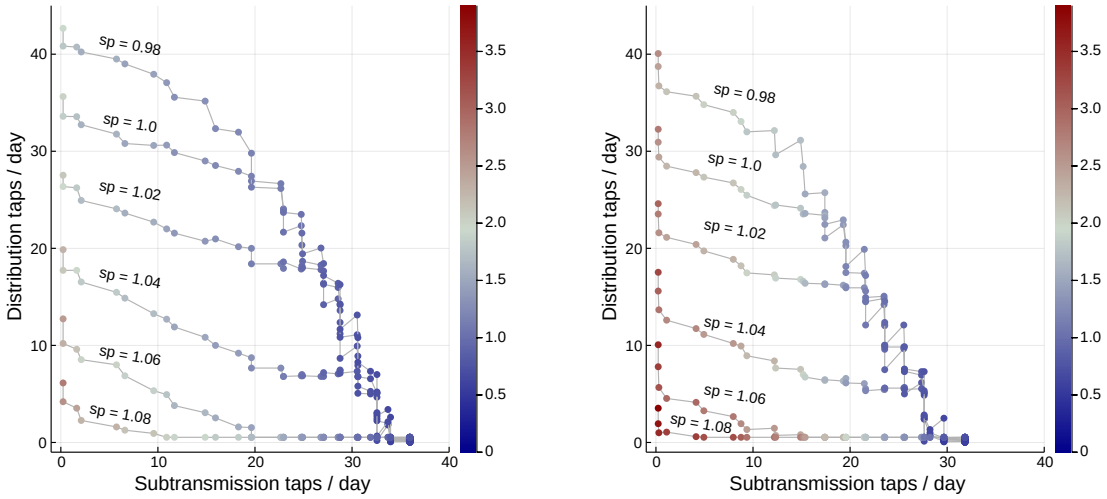
$$V_q = \sum_{t=1}^{t_{\text{end}}} \|\mathbf{V}_t - \mathbf{V}_{\text{nom}}\| \tag{8.3}$$

With this voltage profile metric defined, we turn next to voltage setpoint sensitivity.

In addition to the individual trade-off curves for particular LTC parameter settings shown in Figure 8.3, we also generated families of trade-off curves to illustrate the effects of changing voltage setpoint. The procedure for generating a family of curves is as follows. 1) Let the distribution LTC (modeled with a definite delay) voltage setpoint vary

0.98 pu to 1.08 pu in steps of 0.2 pu. 2) For each setpoint value, generate a trade-off curve of with 50 Q_{max} values distributed evenly over $[0, 0.35]$. 3) Let each point be colored according to V_q, with blue indicating less deviation from the nominal voltage profile, and red indicating more deviation. We followed this procedure to generate two families of curves: one for wind generation, and the other for solar. The results are shown in Figure 8.5, with wind generation on the left and solar on the right. The color of each point represents V_Q, divided by the number of days to obtain the average accumulation of voltage profile deviations each day. Note the general improvement in voltage profile as Q_{max} increases (moving from left to right in each plot). This is as expected: with increased renewable reactive power capability, the distribution LTC is better able to keep its setpoint within the deadband. As we have established previously, the cost due to additional reactive power flow through the network is a greater number of subtransmission LTC tap operations. It is also interesting to note that regardless of voltage setpoint, all trade-off curves converge as Q_{max} increases. By contrast, each curve tapers off to a different number of distribution taps per day with low Q_{max}. This is due to tap saturation. With low Q_{max}, the distribution LTC frequently encounters its tap position limit and remains stuck there, despite the need for further voltage regulation. As voltage setpoint increases, the device encounters its tap limit more frequently, and the number of tap operations decreases (along with voltage profile quality). This explains why V_Q is higher (more voltage profile deviation, redder color) towards the left of each plot. To summarize, each trade-off curve has a region of blue points where both LTCs are able to perform voltage regulation without encountering tap position limits too frequently. The lower the distribution LTC setpoint, the broader this region is. Within the blue region, it is possible to trade off between distribution and subtransmission tap operations with no detrimental effects on voltage profile. Beyond a certain point, any further reduction in distribution LTC tap operations is due to tap saturation, and the voltage profile suffers. It is straightforward to imagine (and generate) similar plots to illustrate the effects of varying subtransmission LTC voltage setpoint, or indeed any other LTC or system parameter. In any case, the intersection of a acceptable voltage profile region and a region of feasible parameter settings restricts the extent to which distribution and subtransmission tap operations may be traded off.

The results generated in our case study suggest great complexity of LTC tapping trade-off phenomenon in general networks. The simulations here have shown that slight changes can significantly alter the shape and viable region of the trade-off curve. Many possibilities beyond those considered here arise in real power systems. The distribution feeder may have its own renewable generation, the subtransmission LTC might serve multiple distribution feeders, or several renewable sources could be connected at the subtransmission level. The



(a) Simulated with wind generation.

(b) Simulated with solar generation.

Figure 8.5: Effects of varying distribution LTC voltage setpoint on tap operations and voltage profile. Simulation was conducted for one month of data (June) at a one-minute time resolution. Each trade-off curve was generated by varying $Q_{\max}$ over $[0, 0.35]$. Points are colored according to V_Q, divided by the number of days (30) to obtain the average voltage profile deviation (in a 2-norm sense) accumulated each day.

need for simulation makes it impractical to simultaneously consider the effects of numerous parameter variations, yet the methodology used to produce plots like those of Figures 8.3 and 8.5 may be used to illuminate the tapping trade-off for one or two parameter variations at a time.

8.5 Conclusion

For many decades load tap changing (LTC) transformers and voltage regulators have been used to keep voltages within bounds throughout sub-transmission and distribution systems. These devices are more than capable of handling the gradual, daily fluctuations of conventional demand. But with the rise of renewable generation, tapping due to load variation is frequently eclipsed by the tapping induced by weather variation. Unfortunately, this problem cannot be solved by simply regulating voltage at renewable nodes. Any benefit to the distribution LTC may come at a cost to the upstream subtransmission LTC. We illustrated this trade-off for two LTC models (definite delay and inverse delay) and two renewable sources (wind and solar). We showed that voltage regulation at the point of interconnection of renewable generation effectively determines how much voltage variability must be compensated by the distribution transformer, and how much is diverted upstream to the

subtransmission transformer.

Tap minimization calls for sophisticated control schemes capable of jointly addressing subtransmission and distribution tap operations. Model Predictive Control (MPC) is well-suited for using continually-updated forecast information to achieve this goal. A differential-algebraic-discrete power system model [134] may be incorporated into an MPC scheme which selects appropriate voltage regulation for renewable nodes and controllable LTCs, and policies for other regulation devices [135]. There are many important details one must consider when designing and implementing such a control scheme. We have shown, for example, that simply switching between LTC models or modifying timing parameters can significantly impact results. Understanding the transformer tapping trade-off is the first step towards more effective voltage control.

CHAPTER 9

Conclusions

This thesis introduced a method for quantifying and managing a transmission network's vulnerability (in terms of transmission line heating) to fluctuations in renewable generation. Another branch of research focused on time series data characteristics and system structure. Electric load data is an important ingredient in the temporal deviation scanning method, but some publicly available load profile datasets exhibit unrealistic characteristics. Time series analysis methods were used to identify these characteristics, and results from signal processing and random matrix theory were used to identify and isolate quantization noise. As with time series datasets, some commonly-used test systems possess unrealistic structural features. Graph-theoretic methods were used to efficiently reveal anomalous connectivity patterns. This led to an exploration of the relationship between system structure and SDP OPF solution performance. Finally, it was shown that voltage regulation at renewable generation sites is not a panacea for voltage magnitude fluctuations due to renewable generation unpredictability. As voltage regulation is increased at renewable generation sites, downstream LTC transformers tap less, but upstream LTC transformers tap more.

Chapter 2 introduced the static deviation scanning problem, which yields a ranked list of renewable generation deviations that drive at least one transmission line to its steady-state flow limit. The method becomes more conservative when decoupled power flow is used (i.e. when non-flat voltage magnitude profiles are accommodated). The inability of static methods to incorporate the time-dependent phenomenon of line failure or dynamic system behavior more generally make them ill-suited for the operating environment. Even so, static deviation scanning provides an interesting testbed for future comparison of power flow approximations and relaxations.

Chapter 3 introduced a temporal deviation scanning framework. By formulating and solving a QCQP as proposed, it is possible to compute the smallest deviation in a wind power forecast that would drive a transmission line to its thermal (temperature) limit over a specified time horizon. The solution algorithm is suitable for large-scale power systems,

and there are a few useful ways to apply the algorithm to address system challenges. Temporal deviation scanning can identify those lines most vulnerable to renewable generation fluctuations. It can also efficiently orchestrate conventional generators to reduce the temperature of a line operating above its limit. In an operational setting, temporal deviation scanning may be used repeatedly to manage system vulnerability to fluctuations, and to keep line temperatures low. Future work should refine the proposed MPC method, especially by justifying or improving the heuristics used. It would also be beneficial to more rigorously study uncertainty in renewable generation and load, and joint uncertainty in both, as they pertain to temporal deviation scanning. Temporal deviation scanning may also prove more valuable in conjunction with dynamic line rating schemes; this merits further consideration. Finally, to ensure the method's viability for realistic power systems, it would be valuable to collaborate with a system operator such as ERCOT in Texas or RTE in France.

Chapter 4 identified distinguishing features of realistic and synthetic electric load data. Six load profile datasets were checked for patterns using cross-correlation analysis and singular value decomposition. These methods help detect whether a load profile is artificial, and whether disaggregation was performed when the full dataset was generated. Individual time series were also considered. We proposed an embedding dimension and eigen-grouping for SSA analysis of load profiles, thereby isolating trends from oscillations and noise. By applying these metrics and analysis methods, one can quickly determine the realism of many load profile datasets. In future work, similar techniques may be applied to other time series data including wind and solar generation.

Chapter 5 considered quantization noise, a nonlinear and irreversible degradation which results from numerical rounding of floating-point data. Having considered the validity of the additive white noise model in the load profile setting, we derived error estimates and bounds. To the author's knowledge, these bounds are the first to accurately predict the singular value "shelf" observed in two popular publicly-available load profile datasets. The additive white noise model also suggests a means of isolating quantization noise. In fact, even large amounts of quantization noise may be cleanly removed if the original load profiles used in disaggregation are known. Literature in signal processing and data science may contain other relevant noise bounds and techniques, calling for future work.

In Chapter 6, a number of topological graph metrics were applied to nearly all publicly-available networks. The analysis revealed several structural anomalies, particularly in PEGASE networks, which likely resulted from network reduction. While some of these undesirable structures had been previously identified, to the author's knowledge this analysis was the first to leverage multiple unweighted graph methods to provide a systematic means

of identifying all of them. The degree of context provided by dozens of publicly available networks is also unprecedented in the literature. A combination of node degree distribution, degree assortativity, and rich-club coefficient metrics is effective for identifying these structures within large graphs in a computationally efficient manner. These metrics may be used to quickly scan new grid models for potential modeling or data issues. The importance of electrical information (as encoded in weighted graphs), along with connections between unweighted and weighted graph properties, may be explored in future work.

Chapter 7 explored the connection between clique structure and decompositions of the semidefinite constraint in the semidefinite relaxation of the optimal power flow problem. While greedy clique merge performs much better than either the full constraint or the fully decomposed constraint, it does not identify the decomposition that minimizes semidefinite solver time. There is a performance gap that suggests the potential for improving SDP OPF computational performance by identifying better constraint decompositions.

Finally, Chapter 8 studied the relationship between voltage control at renewable sites and LTC transformer tap operations. Any benefit provided to the distribution LTC by voltage regulation comes at a cost to the upstream subtransmission LTC. This trade-off was illustrated for two LTC models (definite delay and inverse delay) and two renewable sources (wind and solar). It was shown that voltage regulation at the point of interconnection of renewable generation effectively determines how much voltage variability must be compensated by the distribution transformer, and how much is diverted upstream to the subtransmission transformer. As renewable voltage regulation capability varies, there is a trade-off between subtransmission and distribution LTC tapping. In some regions of this trade-off curve, there are also significant impacts on the system voltage profile. While the LTC tapping trade-off has been discussed in the literature, this analysis provided the most detailed treatment of its relationship to LTC and system parameters, and to the system voltage profile. Minimizing overall tapping calls for sophisticated control schemes capable of jointly addressing subtransmission and distribution tap operations. Model Predictive Control (MPC) is well-suited for using continually-updated forecast information to achieve this goal. There are many important details one must consider when designing and implementing such a control scheme. It was shown, for example, that even a slight change in timing parameters can significantly impact results. Understanding the transformer tapping trade-off is the first step towards more effective voltage control. Based on the work in this chapter, it seems that coordinating LTCs and renewable generators to reduce tap operations is a complex problem whose solution may not scale well to realistic power systems. Nonetheless, future work towards understanding and optimizing around the LTC tapping trade-off should prove valuable.

BIBLIOGRAPHY

[1] M. Chertkov, F. Pan, and M. Stepanov, "Predicting failures in power grids: The case of static overloads," *Smart Grid, IEEE Transactions on*, vol. 2, no. 1, pp. 162–172, March 2011. (pages 2, 10, 11)

[2] M. Chertkov, M. Stepanov, F. Pan, and R. Baldick, "Exact and efficient algorithm to discover extreme stochastic events in wind generation over transmission power grids," in *Proc. 2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 2174–2180. (pages 2, 10, 11, 25)

[3] S. Baghsorkhi and I. Hiskens, "Analysis tools for assessing the impact of wind power on weak grids," in *Systems Conference (SysCon), 2012 IEEE International*, March 2012, pp. 1–8. (pages 2, 10, 12)

[4] C. J. Wallnerström, Y. Huang, and L. Söder, "Impact From Dynamic Line Rating on Wind Power Integration," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 343–350, Jan. 2015. (pages 2, 25)

[5] R. Xiao, Y. Xiang, L. Wang, and K. Xie, "Power System Reliability Evaluation Incorporating Dynamic Thermal Rating and Network Topology Optimization," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6000–6012, Nov. 2018. (pages 2, 25)

[6] M. Nick, O. Alizadeh-Mousavi, R. Cherkaoui, and M. Paolone, "Security Constrained Unit Commitment With Dynamic Thermal Line Rating," *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2014–2025, May 2016. (pages 2, 25)

[7] F. Teng, R. Dupin, A. Michiorri, G. Kariniotakis, Y. Chen, and G. Strbac, "Understanding the Benefits of Dynamic Line Rating Under Multiple Sources of Uncertainty," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3306–3314, May 2018. (pages 2, 25)

[8] Z. Wang, A. Scaglione, and R. J. Thomas, "The Node Degree Distribution in Power Grid and Its Topology Robustness under Random and Selective Node Removals," in *2010 IEEE International Conference on Communications Workshops*, May 2010, pp. 1–5. (pages 3, 104, 106)

[9] P. Hines, S. Blumsack, E. C. Sanchez, and C. Barrows, "The Topological and Electrical Structure of Power Grids," in *2010 43rd Hawaii International Conference on System Sciences*, Jan. 2010, pp. 1–10. (pages 3, 104, 106)

[10] R. Albert, I. Albert, and G. L. Nakarado, "Structural Vulnerability of the North American Power Grid," *Physical Review E*, vol. 69, no. 2, Feb. 2004, arXiv: cond-mat/0401084. (pages 3, 104, 106)

[11] P. Cuffe, "Assortativity Anomalies in a Large Test System," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 4169–4170, Sep. 2016. (pages 3, 104, 107, 109, 110, 112)

[12] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid Structural Characteristics as Validation Criteria for Synthetic Networks," *IEEE Trans. Power Syst.*, vol. PP, no. 99, pp. 1–1, 2016. (pages 3, 104)

[13] E. Cotilla-Sanchez, P. D. H. Hines, C. Barrows, and S. Blumsack, "Comparing the Topological and Electrical Structure of the North American Electric Power Infrastructure," *IEEE Systems Journal*, vol. 6, no. 4, pp. 616–626, Dec. 2012. (pages 3, 104, 106, 107)

[14] A. B. Birchfield, K. M. Gegner, T. Xu, K. S. Shetye, and T. J. Overbye, "Statistical Considerations in the Creation of Realistic Synthetic Power Grids for Geomagnetic Disturbance Studies," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1502–1510, Mar. 2017. (pages 3, 104, 106)

[15] Z. Wang, R. J. Thomas, and A. Scaglione, "Generating Random Topology Power Grids," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, Jan. 2008, pp. 183–183. (pages 3, 104, 106, 107)

[16] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L. DeMarco, "Implementation of a Large-Scale Optimal Power Flow Solver Based on Semidefinite Programming," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3987–3998, Nov. 2013. (pages 3, 105, 115, 116)

[17] H. Li and H. Leite, "Increasing distributed generation using automatic voltage reference setting technique," in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, July 2008, pp. 1–7. (pages 3, 126)

[18] M. Stifter, R. Schwalbe, W. Tremmel, S. Henein, H. Brunner, B. Bletterie, A. Abart, F. Herb, and R. Pointner, "DG DemoNet: Experiences from volt/var control field trials and control algorithm advancements," in *2012 3rd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, October 2012, pp. 1–7. (pages 3, 126, 127)

[19] B. McFetridge, "How to reduce operations on load tap changers at distribution substation transformers," in *2014 IEEE PES T&D Conference and Exposition*, April 2014, pp. 1–5. (pages 3, 126, 127)

[20] Y. Agalgaonkar, B. Pal, and R. Jabr, "Distribution voltage control considering the impact of PV generation on tap changers and autonomous regulators," *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 182–192, January 2014. (pages 3, 126, 127)

[21] Z. Hu, X. Wang, H. Chen, and G. Taylor, "Volt/VAr control in distribution systems using a time-interval based approach," *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 150, no. 5, pp. 548–554, September 2003. (pages 3, 127)

[22] R.-H. Liang and C.-K. Cheng, "Dispatch of main transformer ULTC and capacitors in a distribution system," *IEEE Transactions on Power Delivery*, vol. 16, no. 4, pp. 625–630, October 2001. (pages 3, 127)

[23] Y. Liu, P. Zhang, and X. Qiu, "Optimal reactive power and voltage control for radial distribution system," in *IEEE Power Engineering Society Summer Meeting, 2000*, vol. 1, 2000, pp. 85–90 vol. 1. (pages 3, 127)

[24] J. Kersulis, I. Hiskens, M. Chertkov, S. Backhaus, and D. Bienstock, "Temperature-based instanton analysis: Identifying vulnerability in transmission networks," in *PowerTech, 2015 IEEE Eindhoven*, Jun. 2015. (pages 4, 25)

[25] J. A. Kersulis and I. A. Hiskens, "Renewable voltage regulation and the transformer tapping trade-off," in *2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*. IEEE, 2016, pp. 960–965. (page 5)

[26] J. A. Kersulis, I. A. Hiskens, C. Coffrin, and D. K. Molzahn, "Topological graph metrics for detecting grid anomalies and improving algorithms," in *2018 Power Systems Computation Conference (PSCC)*. IEEE, 2018, pp. 1–7. (page 5)

[27] H. Banakar, N. Alguacil, and F. Galiana, "Electrothermal coordination part I: theory and implementation schemes," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 798–805, May 2005. (pages 7, 24)

[28] C. Coffrin and P. Van Hentenryck, "A Linear-Programming Approximation of AC Power Flows," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 718–734, Nov. 2014. [Online]. Available: http://pubsonline.informs.org/doi/abs/10.1287/ijoc.2014.0594 (pages 12, 20, 104, 108)

[29] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004. (pages 13, 31, 32)

[30] K. C. S. J. Vara Prasad, "Optimal Re-Scheduled Generation using Participation Factors," *International Journal of Computer Applications*, vol. 20, no. 5, pp. 29–32, 2011. (page 14)

[31] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, "Julia: A Fast Dynamic Language for Technical Computing," *arXiv:1209.5145 [cs]*, Sep. 2012. [Online]. Available: http://arxiv.org/abs/1209.5145 (page 19)

[32] C. Coffrin, P. Van Hentenryck, and R. Bent, "Approximating line losses and apparent power in AC power flow linearizations," pp. 1–8. (page 20)

[33] C. Coffrin, H. L. Hijazi, and P. V. Hentenryck, "The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3008–3018, Jul. 2016. (page 20)

[34] ——, "Strengthening the SDP Relaxation of AC Power Flows With Convex Envelopes, Bound Tightening, and Valid Inequalities," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3549–3558, Sep. 2017. (pages 20, 115)

[35] J. Rogers, S. Fink, and K. Porter, "Examples of wind energy curtailment practices," *National Renewable Energy Laboratory*, 2010. (page 25)

[36] B. Parsons, M. Milligan, B. Zavadil, D. Brooks, B. Kirby, K. Dragoon, and J. Caldwell, "Grid impacts of wind power: a summary of recent studies in the united states," *Wind Energy*, vol. 7, no. 2, pp. 87–108, 2004. (page 25)

[37] M. Chertkov, F. Pan, and M. Stepanov, "Predicting failures in power grids: The case of static overloads," *IEEE Transactions on Smart Grid*, vol. 2, no. 1, pp. 162–172, Mar. 2011. (page 25)

[38] J. Cao, W. Du, and H. F. Wang, "Weather-Based Optimal Power Flow With Wind Farms Integration," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3073–3081, Jul. 2016. (page 25)

[39] J. Jiang, Y. Liang, C. Chen, X. Zheng, C. Chuang, and C. Wang, "On Dispatching Line Ampacities of Power Grids Using Weather-Based Conductor Temperature Forecasts," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 406–415, Jan. 2018. (page 26)

[40] D. Douglass, W. Chisholm, G. Davidson, I. Grant, K. Lindsey, M. Lancaster, D. Lawry, T. McCarthy, C. Nascimento, M. Pasha, J. Reding, T. Seppa, J. Toth, and P. Waltz, "Real-Time Overhead Transmission-Line Monitoring for Dynamic Rating," *IEEE Transactions on Power Delivery*, vol. 31, no. 3, pp. 921–927, Jun. 2016. (page 26)

[41] D. L. Alvarez, F. F. d. Silva, E. E. Mombello, C. L. Bak, and J. A. Rosero, "Conductor Temperature Estimation and Prediction at Thermal Transient State in Dynamic Line Rating Application," *IEEE Transactions on Power Delivery*, vol. 33, no. 5, pp. 2236–2245, Oct. 2018. (page 26)

[42] C. R. Black and W. A. Chisholm, "Key Considerations for the Selection of Dynamic Thermal Line Rating Systems," *IEEE Transactions on Power Delivery*, vol. 30, no. 5, pp. 2154–2162, Oct. 2015. (page 26)

[43] B. P. Bhattarai, J. P. Gentle, T. McJunkin, P. J. Hill, K. S. Myers, A. W. Abboud, R. Renwick, and D. Hengst, "Improvement of Transmission Line Ampacity Utilization by Weather-Based Dynamic Line Rating," *IEEE Transactions on Power Delivery*, vol. 33, no. 4, pp. 1853–1863, Aug. 2018. (page 26)

[44] A. Wood and B. Wollenberg, *Power Generation, Operation and Control*, 2nd ed. New York: John Wiley and Sons, Inc., 1996. (pages 26, 31)

[45] "IEEE Standard for Calculating the Current-Temperature Relationship of Bare Overhead Conductors," *IEEE Std 738-2012 (Revision of IEEE Std 738-2006 - Incorporates IEEE Std 738-2012 Cor 1-2013)*, Dec. 2013. (pages 27, 39, 52)

[46] B. Hodge and M. Milligan, "Wind power forecasting error distributions over multiple timescales," in *2011 IEEE Power and Energy Society General Meeting*, Jul. 2011. (page 28)

[47] J. Tastu, P. Pinson, and H. Madsen, "Space-time trajectories of wind power generation: Parameterized precision matrices under a Gaussian copula approach," in *Modeling and Stochastic Learning for Forecasting in High Dimensions*, ser. Lecture Notes in Statistics, X. Brossat, Ed. Springer, 2015, pp. 267–296. (pages 28, 36)

[48] D. Bienstock and A. Michalka, "Polynomial Solvability of Variants of the Trust-region Subproblem," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '14. Portland, Oregon: SIAM, 2014, pp. 380–390. [Online]. Available: http://dl.acm.org/citation.cfm?id=2634074.2634102 (page 30)

[49] O. Mehanna, K. Huang, B. Gopalakrishnan, A. Konar, and N. Sidiropoulos, "Feasible point pursuit and successive approximation of non-convex QCQPs," *IEEE Signal Processing Letters*, vol. PP, no. 99, pp. 1–1, 2014. (page 30)

[50] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996. (page 32)

[51] W. Gander, G. H. Golub, and U. von Matt, "A constrained eigenvalue problem," *Linear Algebra and its Applications*, vol. 114/115, pp. 815–839, 1989. (pages 34, 35, 55)

[52] T. Davis and W. Hager, "Modifying a sparse Cholesky factorization," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 3, pp. 606–627, 1999. (page 36)

[53] L. V. Foster and T. Davis, "Reliable calculation of numerical rank, null space bases, basic solutions and pseudoinverse solutions using suitesparseqr," in *Householder Symposium XVIII on Numerical Linear Algebra*, 2011, p. 79. (page 36)

[54] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 789–821, 1996. (page 37)

[55] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4. (page 37)

[56] P. Cuffe and A. Keane, "Visualizing the Electrical Structure of Power Systems," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2015. (pages 40, 104)

[57] C. Coffrin, D. Gordon, and P. Scott, "NESTA, The NICTA Energy System Test Case Archive," *arXiv:1411.0359 [cs]*, Nov. 2014. (pages 44, 120)

[58] J. Martin and I. Hiskens, "Corrective Model-Predictive Control in Large Electric Power Systems," *IEEE Transactions on Power Systems*, vol. Draft, 2015. (page 46)

[59] M. Almassalkhi and I. Hiskens, "Model-predictive cascade mitigation in electric power systems with storage and renewables – part I: Theory and implementation," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 67–77, January 2015. (pages 46, 53)

[60] B. Otomega, A. Marinakis, M. Glavic, and T. Van Cutsem, "Model Predictive Control to Alleviate Thermal Overloads," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1384–1385, Aug. 2007. (page 46)

[61] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, "The wind integration national dataset (WIND) toolkit," *Applied Energy*, vol. 151, pp. 355–366, August 2015. (pages 46, 129, 131)

[62] L. Wilson, Y. Yang, P. Lu, J. Wang, J. Nielsen-Gammon, N. Smith, and C. Fernandez, "Integrated agricultural information and management system (iAIMS): World climatic data," Jan. 2007. (page 59)

[63] Y. Yang, L. T. Wilson, and J. Wang, "Development of an automated climatic data scraping, filtering and display system," *Computers and Electronics in Agriculture*, vol. 71, no. 1, pp. 77–87, Apr. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169909002348 (page 59)

[64] R. Vautard, P. Yiou, and M. Ghil, "Singular-spectrum analysis: A toolkit for short, noisy chaotic signals," *Physica D: Nonlinear Phenomena*, vol. 58, no. 1, pp. 95–126, Sep. 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016727899290103T (page 80)

[65] P. Bonizzi, J. M. H. Karel, O. Meste, and R. L. M. Peeters, "Singular spectrum decomposition: a new method for time series decomposition," *Advances in Adaptive Data Analysis*, vol. 06, no. 4, p. 1450011, Sep. 2014. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S1793536914500113 (page 80)

[66] A. Miranian, M. Abdollahzade, and H. Hassani, "Day-ahead electricity price analysis and forecasting by singular spectrum analysis," *IET Generation, Transmission & Distribution*, vol. 7, no. 4, pp. 337–346, Apr. 2013. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2012.0263 (page 80)

[67] X. Zhang, J. Wang, and K. Zhang, "Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by cuckoo search algorithm," *Electric Power Systems Research*, vol. 146, pp. 270–285, May 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779617300445 (pages 80, 81)

[68] M. B. Figueiredo, A. de Almeida, and B. Ribeiro, "Wavelet decomposition and singular spectrum analysis for electrical signal denoising." IEEE, Oct. 2011, pp. 3329–3334. [Online]. Available: http://ieeexplore.ieee.org/document/6084183/ (page 80)

[69] M. Lima de Menezes, R. Castro Souza, M. Pessanha, and J. Francisco, "Electricity consumption forecasting using singular spectrum analysis," *DYNA*, vol. 82, no. 190, pp. 138–146, Apr. 2015. [Online]. Available: http://www.scielo.org.co.proxy.lib.umich.edu/scielo.php?script=sci_abstract&pid=S0012-73532015000200017&lng=en&nrm=iso&tlng=en (page 80)

[70] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, December 1974. (page 81)

[71] W. R. Bennett, "Spectra of Quantized Signals," *Bell System Technical Journal*, vol. 27, no. 3, pp. 446–472, Jul. 1948. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01340.x (pages 88, 90, 101)

[72] B. Widrow, "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory," *IRE Transactions on Circuit Theory*, vol. 3, no. 4, pp. 266–276, Dec. 1956. (page 88)

[73] ——, "Statistical analysis of amplitude-quantized sampled-data systems," *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 79, no. 6, pp. 555–568, Jan. 1961. (page 88)

[74] B. Widrow and I. Kollar, *Quantization noise: roundoff error in digital computation, signal processing, control, and communications*. Cambridge University Press, 2008, OCLC: ocn183916250. (page 88)

[75] R. M. Gray, "Quantization noise spectra," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1220–1244, Nov. 1990. (page 88)

[76] B. M. Oliver, J. R. Pierce, and C. E. Shannon, "The Philosophy of PCM," *Proceedings of the IRE*, vol. 36, no. 11, pp. 1324–1331, Nov. 1948. (pages 88, 90)

[77] A. G. Clavier, P. F. Panter, and D. D. Grieg, "Distortion in a Pulse Count Modulation System," *Transactions of the American Institute of Electrical Engineers*, vol. 66, no. 1, pp. 989–1005, Jan. 1947. (page 88)

[78] A. Edelman and N. R. Rao, "Random matrix theory," *Acta Numerica*, vol. 14, pp. 233–297, May 2005. [Online]. Available: https://www.cambridge.org/core/journals/acta-numerica/article/random-matrix-theory/B291B4E6728E10537C2406CE4C341923 (page 89)

[79] S. O'Rourke, V. Vu, and K. Wang, "Random perturbation of low rank matrices: Improving classical bounds," *Linear Algebra and its Applications*, vol. 540,

pp. 26–59, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379517306420 (page 89)

[80] F. Benaych-Georges and R. R. Nadakuditi, "The singular values and vectors of low rank perturbations of large rectangular random matrices," Mar. 2011. [Online]. Available: https://arxiv.org/abs/1103.2221 (page 89)

[81] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998. (page 89)

[82] J. E. Price and J. Goodin, "Reduced network modeling of WECC as a market design prototype," in *2011 IEEE Power and Energy Society General Meeting*, Jul. 2011, pp. 1–6. (pages 90, 91)

[83] P. M. Subcommittee, "IEEE Reliability Test System," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-98, no. 6, pp. 2047–2054, Nov. 1979. (pages 91, 98)

[84] E. R. C. of Texas, "Hourly Load Data Archives," 2018. [Online]. Available: http://www.ercot.com/gridinfo/load/load_hist/index.html (page 91)

[85] C. Barrows, "Reliability Test System - Grid Modernization Lab Consortium," https://github.com/GridMod/RTS-GMLC, 2017. (page 91)

[86] M. Nouri, N. Miki, and N. Nagai, "ARMA model order estimation based on the SVD of the data matrix," *Journal of the Acoustical Society of Japan (E)*, vol. 15, pp. 383–392, Nov. 1994. (pages 93, 98)

[87] S. Seshu and M. B. Reed, *Linear Graphs and Electrical Networks*, first edition ed. Addison-wesley, 1961. (page 104)

[88] G. A. Pagani and M. Aiello, "The Power Grid as a complex network: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 11, pp. 2688–2700, Jun. 2013. (pages 104, 108)

[89] J. Zhao, H. Zhou, B. Chen, and P. Li, "Research on the Structural Characteristics of Transmission Grid Based on Complex Network Theory," *Journal of Applied Mathematics*, vol. 2014, p. e261798, Apr. 2014. (pages 104, 105, 106)

[90] J. Guo, G. Hug, and O. K. Tonguz, "Intelligent Partitioning in Distributed Optimization of Electric Power Systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1249–1258, May 2016. (page 104)

[91] R. J. Sanchez-Garcia, M. Fennelly, S. Norris, N. Wright, G. Niblo, J. Brodzki, and J. W. Bialek, "Hierarchical Spectral Clustering of Power Grids," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2229–2237, Sep. 2014. (page 104)

[92] P. Cuffe, E. S. Marin, and A. Keane, "For Power Systems, Geography Doesn't Matter, But Electrical Structure Does," *IEEE Potentials*, vol. 36, no. 2, pp. 42–46, Mar. 2017. (page 104)

[93] C. Josz, S. Fliscounakis, J. Maeght, and P. Panciatici, "AC Power Flow Data in MATPOWER and QCQP Format: iTesla, RTE Snapshots, and PEGASE," *arXiv:1603.01533 [math]*, Mar. 2016, arXiv: 1603.01533. (pages 104, 108)

[94] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting Sparsity in Semidefinite Programming via Matrix Completion I: General Framework," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 647–674, Jan. 2001. (pages 105, 108, 115)

[95] M. E. J. Newman, "Mixing patterns in networks," *Physical Review E*, vol. 67, no. 2, Feb. 2003. (page 107)

[96] J. J. McAuley, L. d. F. Costa, and T. S. Caetano, "The rich-club phenomenon across complex network hierarchies," *Applied Physics Letters*, vol. 91, no. 8, p. 084103, Aug. 2007, arXiv: physics/0701290. (page 107)

[97] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol. 363, no. 1, pp. 28–42, Oct. 2006. (page 108)

[98] R. Tarjan and M. Yannakakis, "Simple Linear-Time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs, and Selectively Reduce Acyclic Hypergraphs," *SIAM Journal on Computing*, vol. 13, no. 3, pp. 566–579, Aug. 1984. (page 108)

[99] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency Ranking With Respect to Overloads in Very Large Power Systems Taking Into Account Uncertainty, Preventive, and Corrective Actions," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4909–4917, Nov. 2013. (page 108)

[100] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring Network Structure, Dynamics, and Function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, Pasadena, CA, USA, Aug. 2008, pp. 11–15. (page 109)

[101] P. Cuffe, "A Comparison of Malicious Interdiction Strategies Against Electrical Networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 2, pp. 205–217, Jun. 2017. (page 110)

[102] B. C. Lesieutre, D. K. Molzahn, A. R. Borden, and C. L. DeMarco, "Examining the limits of the application of semidefinite programming to power flow problems," in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2011, pp. 1492–1499. (page 115)

[103] J. Lavaei and S. H. Low, "Zero Duality Gap in Optimal Power Flow Problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, Feb. 2012. (page 115)

[104] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed Optimal Power Flow for Smart Microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sep. 2013. (page 115)

[105] R. A. Jabr, "Exploiting Sparsity in SDP Relaxations of the OPF Problem," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1138–1139, May 2012. (page 115)

[106] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz, "Positive definite completions of partial Hermitian matrices," *Linear Algebra and its Applications*, vol. 58, no. Supplement C, pp. 109–124, Apr. 1984. (page 115)

[107] D. Fulkerson and O. Gross, "Incidence matrices and interval graphs," *Pacific Journal of Mathematics*, vol. 15, no. 3, pp. 835–855, Sep. 1965. [Online]. Available: https://msp.org/pjm/1965/15-3/p11.xhtml (page 116)

[108] W. F. T. Fernando L. Alvarado, "Sparsity in Large-Scale Network Computation," *Control and Dynamic Systems*, vol. 41, 1991. (page 117)

[109] W. Tinney, V. Brandwajn, and S. Chan, "Sparse Vector Methods," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-104, no. 2, pp. 295–301, Feb. 1985. (page 117)

[110] W. Tinney and J. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorization," *Proceedings of the IEEE*, vol. 55, no. 11, pp. 1801–1809, Nov. 1967. (pages 117, 119)

[111] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. (page 117)

[112] M. Heathcote, *J & P Transformer Book*, 13rd ed. Newnes, Nov. 2007. (page 125)

[113] E. Camm, M. Behnke, O. Bolado, M. Bollen, M. Bradt, C. Brooks, W. Dilling, M. Edds, W. Hejdak, D. Houseman, S. Klein, F. Li, J. Li, P. Maibach, T. Nicolai, J. Patino, S. Pasupulati, N. Samaan, S. Saylors, T. Siebert, T. Smith, M. Starke, and R. Walling, "Reactive power compensation for wind power plants," in *IEEE Power Energy Society General Meeting, 2009. PES '09*, July 2009, pp. 1–7. (page 125)

[114] R. Walling, R. Saint, R. Dugan, J. Burke, and L. Kojovic, "Summary of distributed resources impact on power delivery systems," *IEEE Transactions on Power Delivery*, vol. 23, no. 3, pp. 1636–1644, July 2008. (pages 125, 126, 127)

[115] "IEEE standard requirements for load tap changers," *IEEE Std C57.131-1995*, pp. 1–46, September 1995. (page 125)

[116] D. Dohnal, "On-load tap-changers for power transformers: A technical digest," *MR Publication, Regensburg*, 2009. (page 126)

[117] J.-H. Choi and S.-I. Moon, "The dead band control of LTC transformer at distribution substation," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 319–326, February 2009. (pages 126, 128)

[118] I.A. Hiskens and D.J Hill, "Dynamic interaction between tapping transformers," in *Proceedings of 11th Power System Computation Conference, Avignon, France*, 1993. (page 126)

[119] N. Yorino, M. Danyoshi, and M. Kitagawa, "Interaction among multiple controls in tap change under load transformers," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 430–436, February 1997. (page 126)

[120] M. S. A. Minhas, J. P. Reynders, and P. J. D. Klerk, "Failures in power system transformers and appropriate monitoring techniques," in *High Voltage Engineering, 1999. Eleventh International Symposium on (Conf. Publ. No. 467)*, vol. 1, 1999, pp. 94–97 vol.1. (page 126)

[121] T. Senjyu, Y. Miyazato, A. Yona, N. Urasaki, and T. Funabashi, "Optimal distribution voltage control and coordination with distributed generation," *IEEE Transactions on Power Delivery*, vol. 23, no. 2, pp. 1236–1242, April 2008. (page 127)

[122] T. Niknam, A. Ranjbar, and A. Shirani, "Impact of distributed generation on volt/var control in distribution networks," in *Power Tech Conference Proceedings, 2003 IEEE Bologna*, vol. 3, June 2003, p. 7 pp. Vol. 3. (page 127)

[123] J.-H. Choi and J.-C. Kim, "Advanced voltage regulation method of power distribution systems interconnected with dispersed storage and generation systems," *IEEE Transactions on Power Delivery*, vol. 16, no. 2, pp. 329–334, April 2001. (page 127)

[124] S. N. Salih, P. Chen, and O. Carlson, "The effect of wind power integration on the frequency of tap changes of a substation transformer," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4320–4327, November 2013. (pages 127, 131)

[125] S. Baghsorkhi and I. Hiskens, "Analysis tools for assessing the impact of wind power on weak grids," pp. 1–8. (page 127)

[126] J. H. Harlow, "Load tap changing control," Beckwith Electric Company, March 1996. (pages 127, 131)

[127] M. Hartung, E.-M. Baerthlein, and A. Panosyan, "Comparative study of tap changer control algorithms for distribution networks with high penetration of renewables," CIRED Workshop, June 2014. (pages 127, 131)

[128] T. Van Cutsem and C. Vournas, *Voltage Stability of Electric Power Systems*. Springer Science & Business Media, November 2007. (pages 128, 131)

[129] J.-Y. Park, S.-R. Nam, and J.-K. Park, "Control of a ULTC considering the dispatch schedule of capacitors in a distribution system," *IEEE Transactions on Power Systems*, vol. 22, no. 2, pp. 755–761, May 2007. (page 128)

[130] M. S. Calovic, "Modeling and analysis of under-load tap-changing transformer control systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-103, no. 7, pp. 1909–1915, July 1984. (page 128)

[131] P. Sauer and M. Pai, "A comparison of discrete vs. continuous dynamic models of tap-changing-under-load transformers," in *Proceedings of Bulk Power System Voltage Phenomena-III: Voltage Stability, Security and Control, Davos, Switzerland*, vol. 11, no. 92, 1994, p. 106. (page 129)

[132] D. Lew, G. Brinkman, and E. Ibanez, "The western wind and solar integration study phase 2," *Contract*, vol. 303, pp. 275–3000, 2013. (pages 129, 131)

[133] M. Larsson and D. Karlss, "Reduce tap changer wear and tear," *T&D World Magazine*, April 1998. [Online]. Available: http://tdworld.com/archive/reduce-tap-changer-wear-and-tear (page 131)

[134] I. Hiskens and M. Pai, "Hybrid systems view of power system modelling," in *The 2000 IEEE International Symposium on Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva*, vol. 2, 2000, pp. 228–231, Vol. 2. (page 137)

[135] D. Opila, A. Zeynu, and I. Hiskens, "Wind farm reactive support and voltage control," in *Bulk Power System Dynamics and Control (iREP) - VIII (iREP), 2010 iREP Symposium*, August 2010, pp. 1–10. (page 137)